AD-A052 338 COMPUTER CORP OF AMERICA CAMBRIDGE MASS F/G 17/2
A PACKET SPEECH MEASUREMENT FACILITY: FINAL TECHNICAL REPORT, 1--ETC(U)
FEB 78 D A LOW MDA903-77-C-0156
UNCLASSIFIED CCA-78-05 NL

1 OF 2
AD
A052338

6

A Packet Speech Measurement Facility: Final Technical Report, 1977

12

B.S.

11 28 Feb 78

12 138p.

14 Technical Report
CCA-78-05
February 28, 1978

10 David A. Low

15 MDA903-77-C-0156

387 285

Computer Corporation of America
575 Technology Square
Cambridge, Massachusetts 02139

12

PSMF Technical Report, 1977

February 28, 1978

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense under Contract No. MDA-903-77-C-0156. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

Table of Contents

A044 469

1. Introduction

The Packet Speech Measurement Facility (PSMF) is a research facility designed to provide members of the Network Secure Communications project with an investigative tool for packetized speech experiments. The PSMF can be invoked to record, playback, and perform measurements on packetized encoded speech streams and may be used to determine the effects of parameter manipulation and network induced perturbations.

This report describes efforts undertaken by the Computer Corporation of America during the first year of PSMF development: December 11, 1976 to December 31, 1977. These efforts have culminated in the implementation and release of a PSMF with operational recording, playback, and measurement functions.

Section 2 details the user's approach to the PSMF. Section 3 is concerned with the process structure and data flow, while Section 4 elucidates details of their implementation. Section 5 outlines PSMF plans for the second year. The Appendices describe some experiments undertaken by PSMF users and demonstrate investigative approaches which will be illuminated by the PSMF in future research.

2. User's Guide to the PSMF

2.1 Overview

The Packet Speech Measurement Facility (PSMF) is an operational recording, playback, and measurement facility designed to provide NSC participants with a tool for packetized speech research. ARPANET access to the PSMF is available using the Network Voice Protocol [COHEN] and an appropriate extension [LOW]. This section is intended to serve as a description of PSMF functions and a guide to their use.

The three critical functions of the PSMF are recording, playback and measurement. The recording function enables a user to designate a PSMF file which will receive a record of negotiation proceedings, voice packets, and special control packets. The playback function provides the means for retrieving part or all of the recorded file. The measurement functions provide tools for analysis of recorded files, and typically create files which can be accessed via the playback function.

A discussion of conventions and an outline of the PSMF extensions are followed by sections which detail each of the PSMF functions. Each function description is accompanied by the protocol necessary to access it, and an example of its use.

2.1.1 Conventions

All communication with the PSMF is via ARPANET single packet messages according to the format described in [COHEN]. The PSMF is host 137 (octal) on the network. Message representations found in the sections will be similar to those used in [COHEN], e.g.

(360) C: 61, 2, <filename-string>, <password-string>

where

(360) is the octal link number.

C: signifies a message sent by the caller; "A" by the answerer.

61,2, are digital representations of 16 bit words.
The header is omitted.

< x > signifies that x is an element to be defined separately. Such elements which are specific to PSMF are

defined below. Others will be found in [COHEN] or [COHEN 2].

2.1.1.1 <String>

A <string> is a sequence of N+1 16 bit words, where the first word is the count of words to follow, i.e., N. Each word contains two 8-bit ASCII bytes, with the byte in the most significant 8 bits of the word preceding the byte in the least significant 8 bits of the word. The string also includes either 1 or 2 NUL (binary zero) terminating bytes, depending on whether there are an odd or even number of text bytes contained in the string, respectively. For example:

```
    "STRING" = 4,ST,RI,NG,00
    ""(null string) = 0 (by definition)
```

A "0" found at the end of a string representation is always meant to signify a null rather than an ASCII "0".

2.1.1.2 <filename-string>

A <filename-string> is a <string> containing at most nine (9) (exclusive of terminating nulls) ASCII characters from the set {A,...,B,0,...,9}, the first character of which must be alphabetic.

2.1.1.3 <Password-string>

A <password-string> is a <string> of arbitrary composition and length (limited by packet size).

2.1.1.4 <field-name>

A <field-name> is a 16 bit word used to identify and retrieve PSMF recorded messages.

<field-name>: = <field-select, field-type>

where <field-select> and <field-type> are the most and least significant 8 bits respectively.

<field-type>'s are

0	-	voice	(labeled implicitly)
1	-	NVP control	(labeled implicitly)
2	-	ASCII text	(labeled explicitly)
3	-	binary	(labeled explicitly)
4	-	graphics	(labeled explicitly)

<field-select>'s are

0	-	select all

and for text,

1	-	author
2	-	sender
3	-	to
4	-	cc
5	-	subject
6	-	date

7 - summary

In addition, non-zero <field-select>'s are produced by some measurement functions to distinguish amongst different categories.

2.1.1.5 <field-name list>

A <field-name list> is a list of N+1 words where the first contains its list count (N) and each of the remaining is a <field-name>.

2.1.1.6 <message-count>

A <message-count> is a signed 32-bit integer.

2.1.1.7 <reason-code>

A <reason-code> is a signed 16-bit integer:

0	-	Unspecified reason
1	-	No access right
3	-	Wrong password
6	-	Illegal <measurement-number>

2.1.1.8 <reason-string>

A <reason-string> is an arbitrary <string>.

2.1.1.9 <measurement-specification>

A <measurement-specification> is a 16-bit word used to identify a PSMF measurement function and select certain input fields.

<measurement-specification>: =

<field-select,measurement-number>

where <field-select> and <measurement-number> are the most and least significant 8 bits respectively.

2.1.1.10 <measurement-modifier>

A <measurement-modifier> is a signed 32 bit integer.

2.1.1.11 <PSMF-integer>

A <PSMF-integer> is a signed 32-bit integer.

2.1.1.12 <PSMF-FP>

A <PSMF-FP> is a 32-bit floating point number in PDP-11 format.

2.1.2 PSMF extension to NVP

The following are PSMF extensions to the NVP. The brief descriptions here will be supplemented in the sections to follow.

60	-	"OK, I AM A PSMF"
61,1	-	"OPEN A FILE TO RECORD"
61,2	-	"OPEN A FILE FOR PLAYBACK"
61,3	-	"OPEN A FILE FOR APPEND"
61,4	-	"RETRIEVE FIELDS"
61,5	-	"FIELD TO BE RECORDED"

61,6	-	"CLOSE FILE"/"END OF FILE"
61,7	-	"DELETE FILE"
61,8	-	"STRING FOR PARSER"
61,9	-	"OPEN OUTPUT MEASUREMENTS"
61,10	-	"OPEN INPUT FOR MEASUREMENT"
61,11	-	"OPEN A FILE TO RECORD (WITH ECHO)"
62,N	-	"FILE OPEN POSITIVE ACKNOWLEDGEMENT"
63,N	-	"FILE OPEN NEGATIVE ACKNOWLEDGEMENT"

2.2 Initial Connection Procedure

The initial connection procedure (ICP) for the PSMF is similar to the standard for NVP.

a.

(377) C : 1, <WHO>, <WHOM>, K

b.

(K) A : 6, L

c.

(L) C : 1, <WHO>, <WHOM>, K

d.

(K) A : 60

In a), b), and c) above, agreement is reached on control links K and L, and implicitly voice links K+1 and L+1. The last response from the PSMF in d) above means "OK, I AM A PSMF" and signifies that the PSMF is awaiting specification of a function.

In addition, the PSMF is designed to respond appropriately to two special NVP messages at any time:

```
        (L)    C : 2       (goodbye)
        (L)    C : 10      (echo)
```

Any messages from the user which are inappropriate under the received context are ignored by the PSMF.

2.3 PSMF Functions

After the initial connection procedure, the PSMF awaits a specification of the desired function. These functions are: recording, playback, measurement, append, delete, and record with echo.

2.3.1 Recording

The PSMF provides a facility for recording a stream of encoded speech messages. Provision is also made to record special PSMF control messages in addition to, or instead of, the speech stream.

The PSMF time-stamps the incoming message stream, classifies it, and later sorts it by user's time stamp. This information is available via measurement functions described in Section 2.3.3.

The recording protocol is:

a. (L) C: 61,1,<filename-string>,<password-string>
This is a command to open a PSMF file with name <filename-string> and access control password <password-string>. If the password string is null, any future access password will suffice.

b. (K) A: 62,1,<filename-string>,<password-string>
This is an acknowledgement from the PSMF. If, however, the <filename-string> is illegal, the PSMF will reply:

(K) A:63,1,<filename-string>,<password-string>,0,0

This is a negative acknowledgement. After issuing this message, the PSMF awaits another function specification as in a) above.

c. (L) C: 61,5,<field-name>, ...

This is a special PSMF control message that can be recorded any time after the file has been opened, even in the midst of a voice stream.

d. (L)(K) NVP voice negotiations as per [COHEN]

These are necessary only if voice messages are to be recorded. Voice negotiations must be instigated by a user's

(L) L: 12,<IM> where IM is non-zero

The PSMF will respond:

(K) A: 13,<IM>,<YM> where <YM>=1

The PSMF will respond affirmatively to all negotiations, except: a negative negotiation response will be given to any maximum message length greater than 976 bits.

e. (L) C: 8

f. (K) A: 6

This is the PSMF's ready reply if information sufficient for playback has been negotiated. (Sample period and samples per parcel.) Otherwise the PSMF responds:

(K) A: 7

g. (L+1) C: encoded voice and/or

(L) C: 61,5,<field-name>, ... PSMF "field to be recorded" messages can be sent imbedded within the voice stream.

h. (L) C: 61,6

or

(L) C: 2

These indicate that the PSMF file is to be closed.

i. (K) A: 61,6,<message-count>

or

(K) A: 61,6,<message-count>

(K) A: 2

These indicate the closure of the PSMF file and the packet count. In the first case, the PSMF awaits another function specification, as immediately after ICP.

The following example may be instructive.

(L) C: 61,1,<3,AL,PH,A0>,<0>

The user wants to open a file named ALPHA for recording. A null password will permit any password for future access.

(K) A: 62,1,<3,AL,PH,A0>,<0>

The PSMF has opened the file.

(L) C: 12,1

The user signifies his intent to open voice negotiations.

(K) A: 13,1,1

The PSMF agrees.

(L) C: 3,3,1,1

The user wants LPC version V1

(K) A: 4,3,1

The PSMF agrees.

(L) C: 8

The user wants to send speech.

(K) A: 6

The PSMF agrees

(L+1) C: encoded speech

(L) C: 61,5,2, "This is a string"

The user sends speech along with a special PSMF control message.

(L) C: 61,6

The user terminates the recording.

(K) A: 61,6,<message-count>

The PSMF closes the file and returns the message count. At this point a function specification is being awaited.

2.3.2 Playback

A previously recorded file can be opened for playback. The user can then specify, by <field-name> those messages he wants the PSMF to retrieve.

The <field-select> subfield of <field-name> is used to qualify the <field-type>, and if 0, signifies that all specimens of a <field-type> are to be returned.

Field-types

0 - all voice messages

1 - all NVP control messages, including those sent by the PSMF during voice negotiation in recording. Useful for debugging.

2 - all PSMF control messages of the form 61,5,2. By convention is "text".

3 - all PSMF control messages of the form 61,5,3. By convention is "binary".

4 - all PSMF control messages of the form 61,5,4. By convention is "graphics".

If voice messages are selected for retrieval, voice negotiation will be instigated and conducted by the PSMF exactly as that which occurred during recording.

The playback protocol is:

a. (L) C: 61,2,<filename-string>,<password-string>

This is a command to open a PSMF file with name <filename-string>, and access password <password-string>. If a null password was specified during recording, this <password-string> is ignored.

b. (K) A: 62,2,<filename-string>,<password-string>

This is an acknowledgement from the PSMF. If, however, no such file exists, or the password is incorrect, the PSMF will reply

(K) A: 63,2,<filename-string>,<password-string>,
<reason-code>, 0

where <reason-code> = 0 if the file could not be opened, and 3 if the wrong password was given.

c. (L) C: 61,4,<field-name list>

The user specifies here the types of messages which are to be retrieved. The order of the list is not important.

If voice messages (<field-name>=0) are specified, the file is scanned for negotiation messages presented during recording (including and terminated by a C:8 (ready?)) and presents them to the user. The PSMF compares the user's response to that which it itself gave during recording. If the response codes are not the same, the PSMF sends a goodbye:

(K) A: 2

Again, if the user specifies the retrieval of voice messages,

d. (K)(L) NVP voice negotiations as per recorded messages.

Playback can be terminated actively by the user in two ways:

e. (L) C: 61,6

If this is sent by the user during or after playback, the PSMF will respond:

f. (K) A: 61,6,<message-count>

The PSMF has closed the file and awaits another function specification as in a) above.

Alternatively the user can send a

g. (L) C: 2

in which case the PSMF replies

h. (K) A: 61,6,<message-count>

(K) A: 2

Playback will be also terminated by the PSMF when no more messages are to be retrieved. In this case, however, the playback file will not have been closed, and PSMF will be awaiting another

<field-name list> just as in c) above. This is the only case in the PSMF in which a file remains open after the PSMF sends an "EOF" message.

It should be noted that playback of control messages on the control link is not timed, but occurs as they are found in the playback file. If the user finds himself in danger of running out of buffer space, he can send a

i. (L) C: 7

meaning he is unready to accept any more data. The PSMF will suspend further transmission until receipt of

j. (L) C: 6

The following example illustrates retrieval of information recorded in the example in Section 2.3.1.

(L) C: 61,2,<3,AL,PH,A0>,<5,PA,SS,WO,RD,00>

The user wants to open the file ALPHA for playback. Because this file was recorded with a null password, any access password will be ignored.

(K) A: 62,2,<3,AL,PH,A0>,(5,PA,SS,WO,RD,00>

The PSMF has opened the file for playback and is awaiting a <field-name list>

(L) C: 61,4,1,2

The user specifies retrieval of <field-type> = 2.

(K) A: 61,5,2, "This is a string"

(K) A: 61,6,<01>

The first message from the PSMF was the only "2" PSMF message recorded by the user. The second is an "EOF" message with the count=1.

(L) C: 61,4,2,2,0

This time the user requests playback of both type "2" and "0" (voice) messages.

(K) A: 12,1

The PSMF signals its intent to conduct voice negotiations by requesting to be negotiation master (even if it already is by default).

(L) C: 13,1,1

The user agrees.

(K) A: 3,3,1,1

The PSMF (mimicing the recording session) wants LPC version V1.

(L) C: 4,3,1

The user agrees.

(K) A: 8

The PSMF asks if the user is ready.

(L) C: 6

The user agrees.

(K+1) A: encoded speech

(K) A: 61,5,2, "This is a string"

The PSMF "2" message is sent back at the point it was imbedded in the voice stream.

(K) A: 61,6,<message-count>

The PSMF has retrieved all specified messages, and awaits another <field-type list>, a PSMF "close file", or a "goodbye" message.

2.3.3 Measurements

A previously recorded file can be opened for measurement. The user can then specify the file which will receive the measurement data, and the measurement function. On completion of the measurement, the resulting file (consisting of 61,5,3 messages) can be played back.

The file created during a measurement function may itself be subjected to further measurement. In this way, for instance, a file of relative delay times might be created followed by another measurement creating a file of its histogram.

A user signals his intent to perform a measurement by:

a. (L) C: 61,10,<filename-string>,<password-string>

This is a command to open a previously created PSMF file for measurement. If the name and password are valid, the PSMF will reply:

b. (K) A: 62,10,<filename-string>,<password-string>

The user then has to specify the output file and measurement specification:

c. (L) C: 61,9,<filename-string>,<password-string>,

```
                <measurement-specification>,
                [<measurement-modifier>,
                 <measurement-modifier>]
```

The PSMF opens an output file with name <filename-string> and password <password-string>. The measurement specified in the <measurement number> subfield of the <measurement-specification> is performed on all input records with the same <field-select> subfield.

Measurement numbers

0 - measurement of relative delays experienced by voice packets.

1 - classification of voice packets into missing, duplicate, or out of order.

2 - measurement of periods of silence and speech.

3 - classification of PSMF records into <field-names>.

4 - computation of mean and standard deviation.

5 - construction of a histogram.

If an invalid <measurement-specification> is used, the PSMF will respond

```
d.  (K)  A: 63,9,<filename-string>,<password-string>,
                   <measurement-specification>,6,0
```

At this point, the PSMF is awaiting another function specification as in a) above.

2.3.3.1 Measurements of Relative Delays

The absence of an absolute time standard means that delay times have to be measured relative to some standard. The PSMF relative delay function produces a file of delays relative to that experienced by the first voice packet:

```
  relative delay = (PSMF time stamp for this message -
                   PSMF time stamp for first message)
                   -
                   (user's  time stamp for this message -
                   user's time stamp for first message)
```

A record of the form

61,5,3,<PSMF-integer>

is created for every input voice packet, including the first. The <PSMF-integer> is in milliseconds.

The user specifies this measurement as

c. (L) C: 61,9,<filename-string>,<password-string>,0

The <field-select> subfield of the <measurement-specification> is ignored; all voice packets are subject to measurement.

The PSMF replies

d. (K) A: 62,9,<filename-string>,<password-string>,0

and when finished,

e. (K) A: 61,6,<message-count>

The output file can later be retrieved via playback, or itself subjected to further measurements.

An example follows:

(L) C: 61,10,<3,AL,PH,A0>,<0>

Here the user signals his intent to perform measurements on the file ALPHA. The password is null.

(K) A: 62,10,<3,AL,PH,A0>,<0>

The PSMF accedes.

(L) C: 61,9,<3,BE,TA,00>,<2,PW,00>,0

The user wants the output from the measurement function to go into a new file "BETA" with password "PW". He specifies measurement #0, or relative delays.

(K) A: 62,9,<3,BE,TA,00>,<2,PW,00>,0

A: 61,6<message-count>

The PSMF agrees, performs the measurements, records them, and closes the file. The <message-count> is the count of output records. The measurements are stored as a PSMF file consisting of a series of

61,5,3,<PSMF-integer>

messages. The PSMF is now awaiting another function specification.

2.3.3.2 Missing, Duplicate, or Out of Order

Some characteristics of voice stream integrity can be measured. The possible presence of embedded control messages complicates things slightly, but it is usually possible to count the number of missing, duplicate, or out of order packets.

Measurement function #1 does that, classifying all received messages (including control) into:

3 - missing (inferred when an expected time stamp is not found, and the subsequent (by user's time stamp) does not have a "we skipped parcels" bit on.)

2 - Duplicate (inferred when a message was not linked into the user's time stamp

sequence during the post-recording sort.)

1 - Out of order (inferred when the above mentioned sort had to link to a previous (by PSMF time stamp) message).

0 - other

The above classifications are considered to be mutually exclusive, except that an "out of order " packet may engender a "missing" message as well. The output records are all of the form:

61,5,3,<PSMF-integer>

where

<PSMF-integer> is an integer from 0 to 3.

The user requests this measurement as

c. (L) C: 61,9<filename-string>,<password-string>,1

The <field-select> subfield of the <measurement-specification> is ignored; all PSMF recorded packets are subject to measurement.

The PSMF replies:

d. (K) A: 62,9,<filename-string>,<password-string>,1

and when finished,

e. (K) A: 61,6,<message-count>

The example in Section 2.3.3.1 serves to illustrate the function as well. The <measurement-number> is, of course, 1.

2.3.3.3 Periods of Speech and Silence

This measurement function scans the encoded voice stream. At every occurrence of a "we skipped parcels" bit in an input voice message, two output messages are created:

i) 61,5,403,<PSMF-integer>

ii) 61,5,1003,<PSMF-integer>

The first is a <field-type>=3, <field-select>=1, <PSMF-integer>=duration of last speech segment in milliseconds. The second is a <field-type>=3, <field-select>=2, <PSMF-integer>=duration of this silence period in milliseconds (up to 65 seconds).

These two types of records can be selectively accessed for playback, or for further measurement functions of histograms and mean-standard deviation.

The user specifies this measurement as

c. (L) C: 61,9,<filename-string>,<password-string>,2

The <field-select> subfield of the <measurement-specification> is ignored; all voice packets are subject to measurement. This function does produce, however, records of two distinct <field-select> varieties.

The PSMF replies:

d. (K) A: 62,9,<filename-string>,<password-string>,2

and, when finished,

e. (K) A: 61,6,<message-count>

The example in Section 2.3.3.1 serves to illustrate this function as well, with the <measurement-number> being 2.

2.3.3.4 Types of message

This function scans the entire input PSMF file and classifies each packet into field types:

0	-	voice message
1	-	NVP control message (either of user or PSMF origin)
2 thru 4	-	PSMF control messages of the form 61,5, <field-select,x>

A

61,5,3,<PSMF-integer>

is created for each input record. <PSMF-integer> ranges from 0-4.

The user specifies this measurement as

c. (L) C: 61,9,<filename-string>,<password-string>,3

The <field-select> subfield of the <measurement-specification> is ignored; all PSMF records are subject to measurement.

The example in Section 2.3.3.1 serves to illustrate this function as well, with the <measurement-number> being 3.

2.3.3.5 Mean and Standard Deviation

This is the first of the two measurement functions which normally use a previously created measurement file as input.

All input records of the form:

61,5,<field-select,3>,<PSMF-integer>

where <field-select> is the same as that in the <measurement-specification>, are scanned. Three output records are created of the form

61,5,3,<PSMF-FP>

where <PSMF-FP> are PDP-11 floating point representations of the count N, mean M and standard deviation estimate S.

M =(Sum(x))/N

```
S = SQRT ((  (x-M)**2)/(N-1))
```

The user specifies this measurement as

c. (L) C: 61,9,<filename-string>,<password-string>,
<field-select,4>

The <field-select> subfield of the <measurement-specification> is used to select input records. This could be used, for instance, to select results from a period of speech or silence measurement.

The PSMF replies:

d. (K) A: 62,9,<filename-string>,<password-string>,4
and when finished,

e. (K) A: 61,6,<message-count>
where <message-count>=3.

2.3.3.6 Histogram

This measurement function normally uses a previously created measurement file as input. All input records of the form

```
     61,5,<field-select,3>,<PSMF-integer>
```

where <field-select> is the same as that in the <measurement-specification>, are scanned. 128 output records are created of the form:

```
     61,5,3 <PSMF-integer>
```

where <PSMF-integer> is the count in the histogram bin as defined below.

The user invokes this measurement as

```
c.   (L)  C: 61,9,<filename-string>,<password-string>,
                <field-select,5>,<measurement-modifier#1>,
                <measurement-modifier#2>
```

where <measurement-modifier#1> is the low bin boundary of the histogram (all measurements below are put into this bin) and <measurement-modifier#2> is the bin interval. The high bin boundary is defined as low bin boundary + 128 * bin interval. All measurements above are put into the high bin.

The PSMF replies:

```
d.    (K)  A: 62,9,  etc.

           and when finished:

e.    (K)  A: 61,6,<message-count>

           where <message-count>=128.
```

The example below will illustrate the use of this function:

First the user performs a measurement of periods of speech and silence on PSMF file ALPHA.

```
(L)(K)  Initial connection procedure
(L)     C: 61,10,<3,AL,PH,A0>,<0>
(K)     A: 62,10,<3,AL,PH,A0>,<0>
(L)     C: 61,9,<3,BE,TA,00>,<2,PW,00>,2
(K)     A: 62,9,<3,BE,TA,00>,<2,PW,00>,2
(K)     A: 61,6,<message-count>
```

Then the user specifies file BETA as input file to the histogram function:

```
(L)     C: 61,10,<3,BE,TA,00>,<2,PW,00>
(K)     A: 62,10,<3,BE,TA,00>,<2,PW,00>
```

and specifies file GAMMA with password OK as output file.

```
(L)     C: 61,9,<3,GA,MM,A0>,<2,OK,00>,<1,5>,
           <0>, <1000>
```

The user has specified measurement 5 (histogram) on field-select 1 (periods of speech) with low boundary 0 and increment 1000 (=1 second).

(K) A: 62,9,<3,GA,MM,A0>,<2,OK,00>,<1,5>,<0>,<1000>

The PSMF goes to it

(K) A: 61,6,<128>

and signals when through.

Finally, the user can playback the results of this histogram:

(L) C: 61,2,<3,GA,MM,A0>,<2,OK,00>

(K) A: 62,2,<3,GA,MM,A0>,<2,OK,00>

(L) C: 61,4,1,3

The user *wants to receive all*
61,5,3, messages.

(K) A: plays back all 128 messages

(K) A: 61,6, <128>

2.3.4 Appending

Data can be appended to a previously existing PSMF file. (If voice is to be appended, the very first segment of the file must contain voice.) Each segment retains some identity, in order to re-synchronize playback and some measurement functions, but cannot be selectively accessed. Only the site (HOST/IMP/EXTENSION) which created the original file is permitted to append.

The appending protocol is:

a. (L) C: 61,3,<filename-string>,<password-string>
   This is a command to append to a previously existing file. If the name and password are valid, the PSMF responds:

b. (K) A: 62,3,<filename-string>,<password-string>
   At this point the user can record information as in Section 2.3.1. If the user wishes to append a voice segment (the file must already contain voice information), he signifies this by requesting that the PSMF become negotiation master.

c. (L) C: 12,0

d. (K) A: 12,0,1

which the PSMF agrees to.

The PSMF then conducts voice negotiations as per those already stored (as in playback). The final "8" = "ready?" message is converted to a "6" = "ready!" message and the PSMF awaits information to be recorded. The following example may illustrate this procedure. Refer to the example of 3.1.

(L) C: 61,3,<3,AL,PH,AO>,<0>

The user wants to append to the file named ALPHA. Since ALPHA was recorded with a null password, any password, including the null above, can be used for access.

(K) A: 62,3, <3,AL,PH,AO>,<0>

The PSMF has opened the file and is ready to append.

(L) C: 61,5,2, "This is text"

The user can append textual data.

(L) C: 12,0

The user signals an intent to append voice.

(K) A: 13,0,1

The PSMF agrees.

(K) A: 3,3,1,1

The PSMF suggests LPC version V1.

(L) C: 4,3,1

The user agrees.

(K) A: 6

The PSMF says it is ready.

(L)+(L+1) C: encoded speech and special PSMF control messages.

(L) C: 61,6

The user terminates appending.

(K) A: 61,6,<message-count>

The PSMF closes the file and returns the message count of the entire (not just the appendage) file. At this point a function specification is being expected.

2.3.5 Deleting

A PSMF file can be deleted by the creation site (same HOST/IMP/EXTENSION). This can be done to get access to a previously created version (which may have, unfortunately, already been purged at the PSMF site), or to free disk resources.

The protocol for deletion is:

```
  (L)  C: 61,7,<filename-string>,<password-string>
  (K)  A: 62,7,<filename-string>,<password-string>
```

After this acknowledgement the PSMF awaits a function specification.

2.3.6 Recording with Echo

Provision has been made to permit PSMF echoing of voice messages and PSMF "61,5" messages ("field to be recorded") at the time of recording. In every respect, this function is the same as the recording function of Section 2.3.1, except that the data is copied back to the user immediately before it is recorded.

The protocol for recording with echo is:

```
  (L) C: 61,11,<filename-string>,<password-string>
  (K) A: 62,11,<filename-string>,<password-string>
```

Again, further interaction with this function is identical to that described in Section 2.3.1.

2.3.7 NVCP Recording

In order to accommodate NSC experimenters who have no PSMF extension implementation, and to allow transcription of message streams in a conversational environment, provision has been made to permit recording of conferences carried out under the Network Voice Conferencing Protocol (NVCP) [COHEN 2]. Such a recording results in a standard PSMF file, with each participant's speech segment being treated as an appended file (see Section 2.3.4).

This facility is not strictly a PSMF function: the initial connection procedure is different and only a recording function is provided. Any playback or measurement of the recorded file, however, is conducted using standard PSMF facilities.

Again, the ICP differs from that of the PSMF:

The user sends to the PSMF a

```
(377) C:42,<chairman>,<participant>,<filename-string>, <password-string
```

This is essentially a "PLEASE JOIN MY CONFERENCE" message, as described in [COHEN 2], with the addition of a PSMF file specification.

From this point on, the protocol is standard NVCP. The PSMF sends a message to the chairman: "A PARTICIPANT WANTS TO JOIN YOUR CONFERENCE"

```
(377) A:33,<chairman>,<user>, K
   etc.
```

The record which is made of voice negotiations is compatible with PSMF playback needs. This permits the PSMF playback and measurement functions to accommodate themselves to the timestamp discontinuity between different users' speech segments.

2.3.8 Non-PSMF Echo Facility

For the convenience of NSC experimenters who need a remote loop-back, a non-PSMF echo facility has been installed at the PSMF site. This facility will, after negotiation completion, simply echo all voice packets back to the user. No use is made of the PSMF extension to the NVP.

The echo facility is invoked when the user addresses the user site "extension" 0 during ICP, i.e.:

a. (377) C: 1, <WHO>, <WHOM>, K

where the format of <WHOM> is

(HOST/IMP/EXTENSION=0)

The echo facility replies:

b. (K) A: 6, L

as in the standard NVP initial connection.

The caller re-states his connection:

c. (L) C: 1, <WHO>, <WHOM>, K

At this point, the echo facility suggests negotiation parameters:

d. (K) A: 3,3,3,1,2,3

asking the user if he can do versions 1, 2 or 3. If the reply is negative, the echo facility suggests that the user become negotiation master. Otherwise, a maximum packet size of 976 bits is negotiated, ready signals are exchanged, and the echoing of voice packets can commence. This facility terminates itself on receipt of a

(L) C: 2

goodbye message.

2.4 Parsing Facility

A facility is available to ease the invocation of PSMF functions described in Section 2.3. When the PSMF is awaiting a function specification, as in immediately after the Initial Connection Procedure, a message of the form

```
(L) C: 61,8, <string>
```

may be parsed into one or two PSMF function commands before being passed to the PSMF. All key words described below can be abbreviated to three or more characters. Numbers are considered to be octal unless terminated by a".". All filename/password elements must begin with a non-numeric, contain a "/", and have no imbedded blanks.

PSMF responses˜to commands produced by the parser are addressed to the user, as though they originated from him.

2.4.1 Recording

If the keyword "RECORD" is the first, non-blank element of the <string>, a filename/password (where "/" is a required character) is parsed next and a 61,1 message as described in Section 2.3.1 is sent to the PSMF. E.g. a:

```
  (L)  C:  61,8, <RECORD TEST/PW>
```

is converted to a

```
  (L)  C:  61,1, <TEST>, <PW>
```

2.4.2 Playback

If the keyword "PLAYBACK" is the first non-blank element of the <string>, a filename/password (where "/" is a required character) is parsed next, and a 61,2 message as described in section 2.3.2 is sent to the PSMF.

If the character ">" is present, the parsing continues to produce a 61,4 message and a field name list. The keywords for <field-type> are "VOICE", "CONTROL", "TEXT", "BINARY", "GRAPHICS" and "ALL" (which is everything except "CONTROL"). <Field-type> keywords may optimally be followed by a "." and a <field-select>. A <field-select> may be a number, or one of the keywords "ALL", "AUTHOR", "SENDER", "TO", "CC", "SUBJECT" or "DATE".

For example:

```
  (L)  C:  61,8, <PLAYBACK TEST/PW>
```

is converted to

```
  (L)  C:  61,2, <TEST>, <PW>
```

while

```
(L)  C:  61,8, <PLAYBACK TEST/PW ">" TEXT.SUBJECT,
                    BINARY.2>
```

is converted to

```
(L)  C:  61,2, <TEST>, <PW>
(L)  C:  61,4,2, <5,2>, <2,3>
```

2.4.3 Measurements

If the keyword "MEASURE" is the first non-blank element of the <string>, a filename/password (where "/" is a required character) is parsed next, and a 61,10 message as described in Section 2.3.3 is sent to the PSMF.

If the character ">" is present, the parsing continues to produce a 61,9 message and measurement specification. The first element parsed here must be the output filename/password. The next element should be one of the keywords: "DELAY", "MISSING", "PERIODS", "TYPES", "MEAN" AND "HISTOGRAM". These keywords may be followed by a list of <measurement.modifier>'s enclosed in parenthesis.

For example:

```
(L)  C:  61,8,<MEASURE TEST/PW>
```

is converted to

```
  (L)  C:  61,10,<TEST>, <PW>
```

while

```
  (L)  C:  61,8, <MEASURE TEST/PW > OUTPUT/PW
                  HISTOGRAM (-1000.,100.) >
```

is converted to

```
  (L)  C:  61,10, <TEST>, <PW>
  (L)  C:  61,9,  <OUTPUT>,<PW>,<5>,<27777776030>,
                  <144>
```

2.4.4 Appending

If the keyword "APPEND" is the first non-blank element of the <string>, a 61,3 message is produced similarly to that for "RECORD" in 2.4.1 above.

2.4.5 Deleting

If the keyword "DELETE" is the first non-blank element of the <string>, a 61,7 message is produced similarly to that for "RECORD" in 2.4.1 above.

2.4.6 Recording With Echo

If the keyword "ECHO" is the first non-blank element of the <string>, a 61,11 message is produced similarly to that for "RECORD" in 2.4.1 above.

3. Process Structure and Data Flow

The Digital Equipment Corporation RSX-11M operating system was chosen as the vehicle for development and implementation of the PSMF. Reasons for this were:

a. A development machine under RSX-11M was available.

b. RSX-11M has adequate software development tools.

c. A well developed file system is offered.

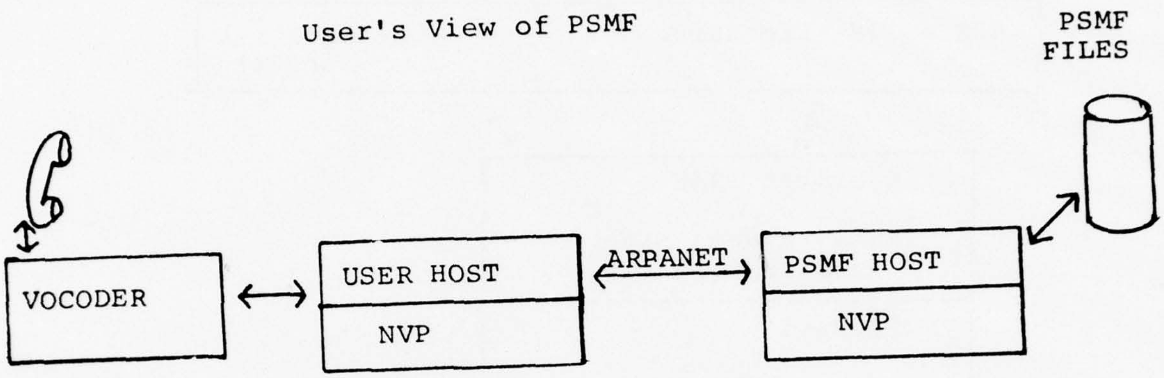
d. RSX-11M provides executive services for scheduling, inter-process communications, timing, I/O, and dynamic memory allocation and mapping.

With these facilities available, it was decided to design the PSMF as a set of functionally disjoint processes. These processes would communicate via executive directives and a common buffer area, and would share the services of a set of library routines.

The PSMF user is, of course, unaware of these components. His view of the system might be as illustrated in Figure 3.1.

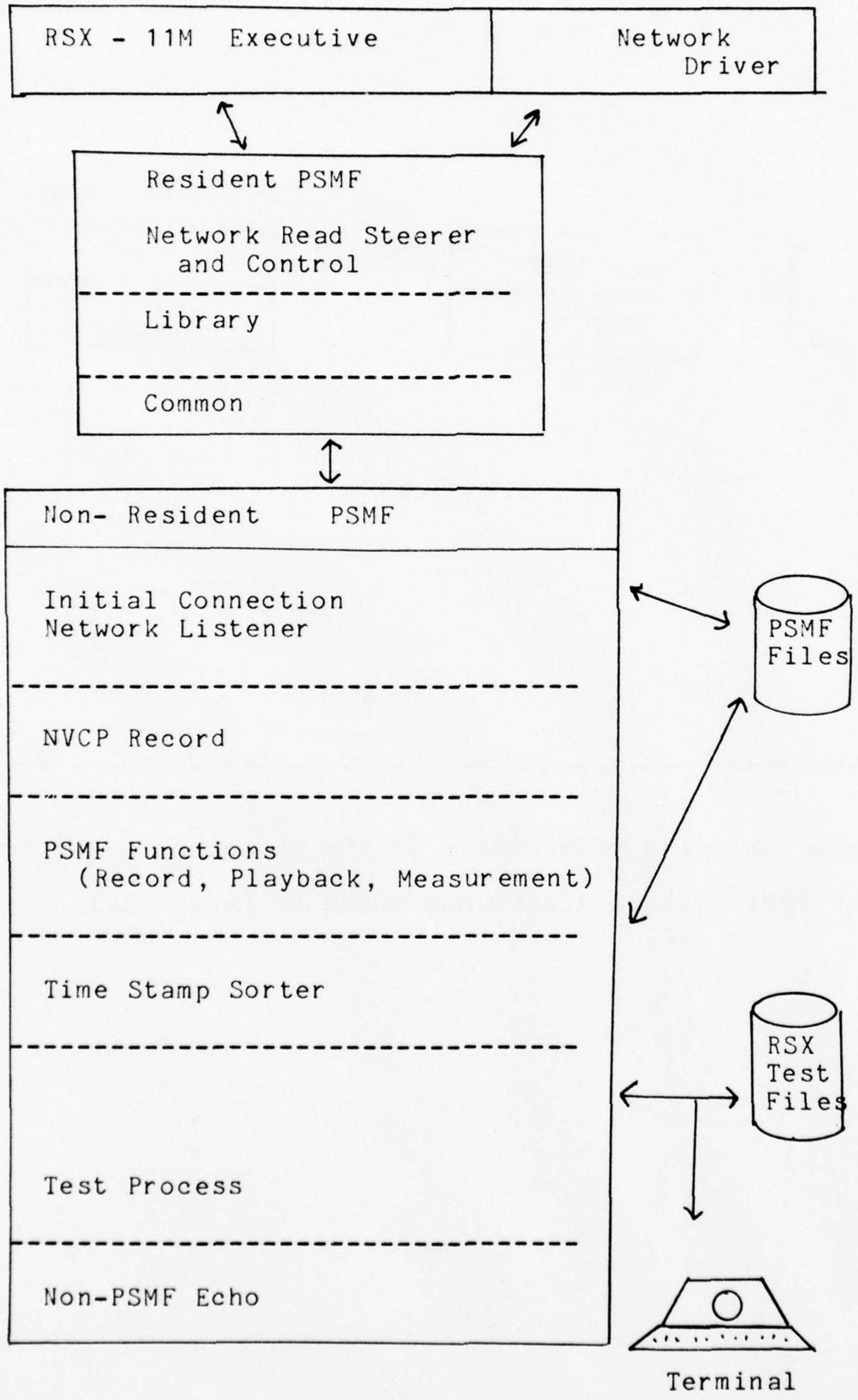
Figure 3.1

User's View of PSMF



A more detailed examination of the structure to be found at the PSMF host is diagrammed below in figure 3.2.

Figure 3.2


RSX - 11M Executive
Network Driver
Resident PSMF
Network Read Steerer and Control
Library
Common
Non- Resident PSMF
Initial Connection Network Listener
NVCP Record
PSMF Functions (Record, Playback, Measurement)
Time Stamp Sorter
Test Process
Non-PSMF Echo
PSMF Files
RSX Test Files
Terminal

3.1 Process Structure

3.1.1 RSX-11M Executive and Network Driver

The executive provides a wide variety of management and communications services. Explicit invocations of executive services are excluded from the PSMF Function process itself (in order to facilitate operating system conversion), but are included in the Library routines, Control process, and Test process.

The network driver provides an interface between PSMF processes and Arpanet. By convention, network reads are requested only by the Control and Network Read process, via the mediation of Library routines. Network writes can be requested by any process, again through a Library routine. The executive manages I/O queuing.

3.1.2 Library Routines

These are a set of re-entrant and position independent routines which occupy part of the address space of all PSMF processes. They incorporate functions common to all processes, and when structured as a Library, serve to reduce total core space requirements, and to isolate explicit references to executive services. RSX-11M requires that a Library be resident.

3.1.3 Common Area

This is a buffer or data area which occupies part of the address space of all PSMF processes, facilitating interprocess communication. RSX-11M permits such an area to be dynamically created.

3.1.4 Control and Network Read Process

This process is initiated from an RSX-11M terminal. Its functions are to:

a. Initialize PSMF data structures.

b. Initiate a process which listens to the NVP initial connection link.

c. Make judgments as to the disposition of all network reads.

d. Communicate to processes, via global semaphores, the results of any function requiring a network read. This process will often be referred to as the "Control" process.

3.1.5 Initial Connection Network Listener Process

This process is initiated by the Control process (3.1.4). Its function is to listen to NVP link 377 for a protocol sequence indicating a user's request for a PSMF function. If this sequence is received, the PSMF Function process is initiated.

3.1.6 PSMF Function Process

This process is initiated by the Listener process (3.1.5) when a user requests a PSMF function, and terminates itself on receipt of an NVP "goodbye" message. It incorporates all the NVP and PSMF extensions necessary to communicate with the user.

The principal PSMF functions are Recording, Playback, and Measurement. Each of these requires access to PSMF disk files. Such access is mediated by a group of routines which specialize RSX file manipulation to the PSMF's needs.

It may be possible that the PSMF Function process initiates asynchronous processes on its own. The Recording function, for instance, may initiate a process which performs a sort on user's voice time stamps.

3.1.7 Test Process

This process is initiated from an RSX-11M terminal. Its function is to simulate user access to the PSMF. As a PSMF process in its own right, it conveniently provides a test of the multi-user capabilities of the design.

3.2 Data Flow

Figures 3.3 through 3.6 illustrate data flow amongst the various processes.

When network data is received on link 377, the Control process steers it to the Initial Connection Process, as shown in Figure 3.3. If the data is of proper format, the Initial Connection Process will start another process (such as the PSMF process, as illustrated in Figure 3.4) which then operates independently. The new process initiates network writes on its own, while receiving network data steered by the Control Process.

Figure 3.3

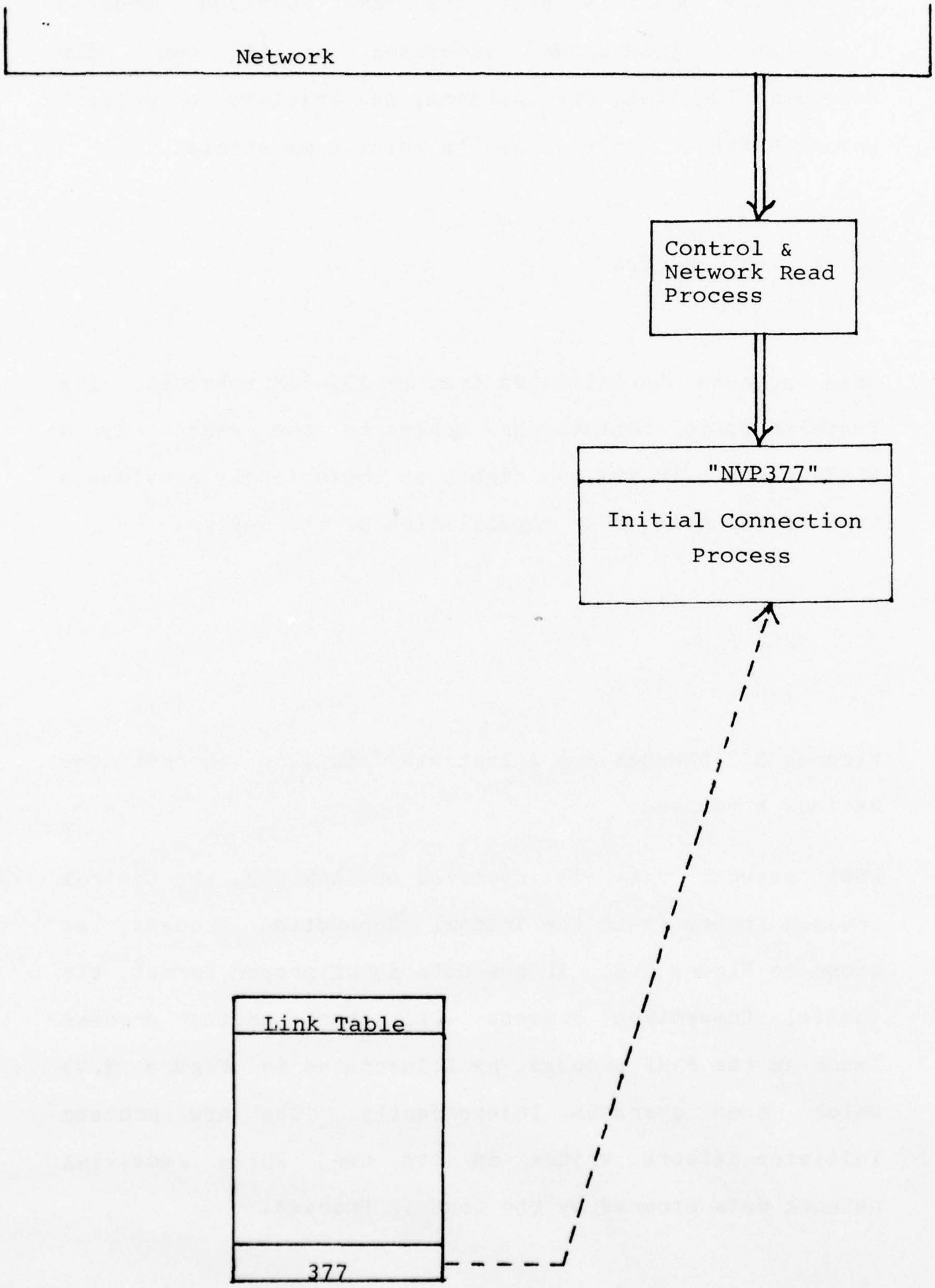

Network
Control & Network Read Process
"NVP377"
Initial Connection Process
Link Table
377

Figure 3.4

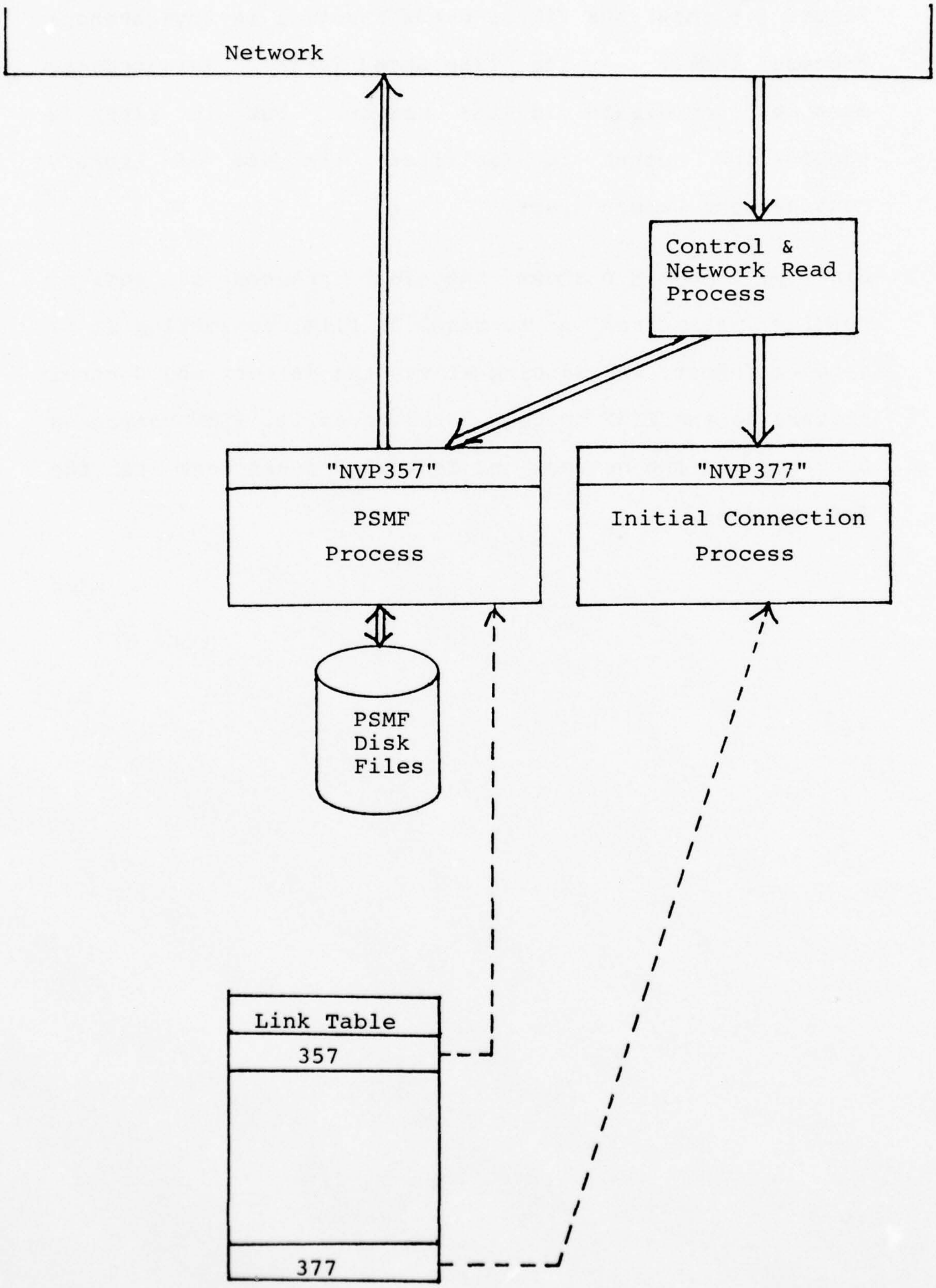

Network
Control &
Network Read
Process
"NVP357"
PSMF
Process
"NVP377"
Initial Connection
Process
PSMF
Disk
Files
Link Table
357
377

Figure 3.5 shows the PSMF process starting an asynchronous process itself -- the Time Stamp Sorter. This process does not communicate via the network, but is given a pseudo-link number to facilitate its use of Library routines and Common space.

Finally, Figure 3.6 shows the Test process in action: reading data from a terminal or file, converting it to network format, and sending it via the network and Control process to the PSMF process. Reciprocally, PSMF responses are sent via the network and Control Process back to the Test process.

Figure 3.5

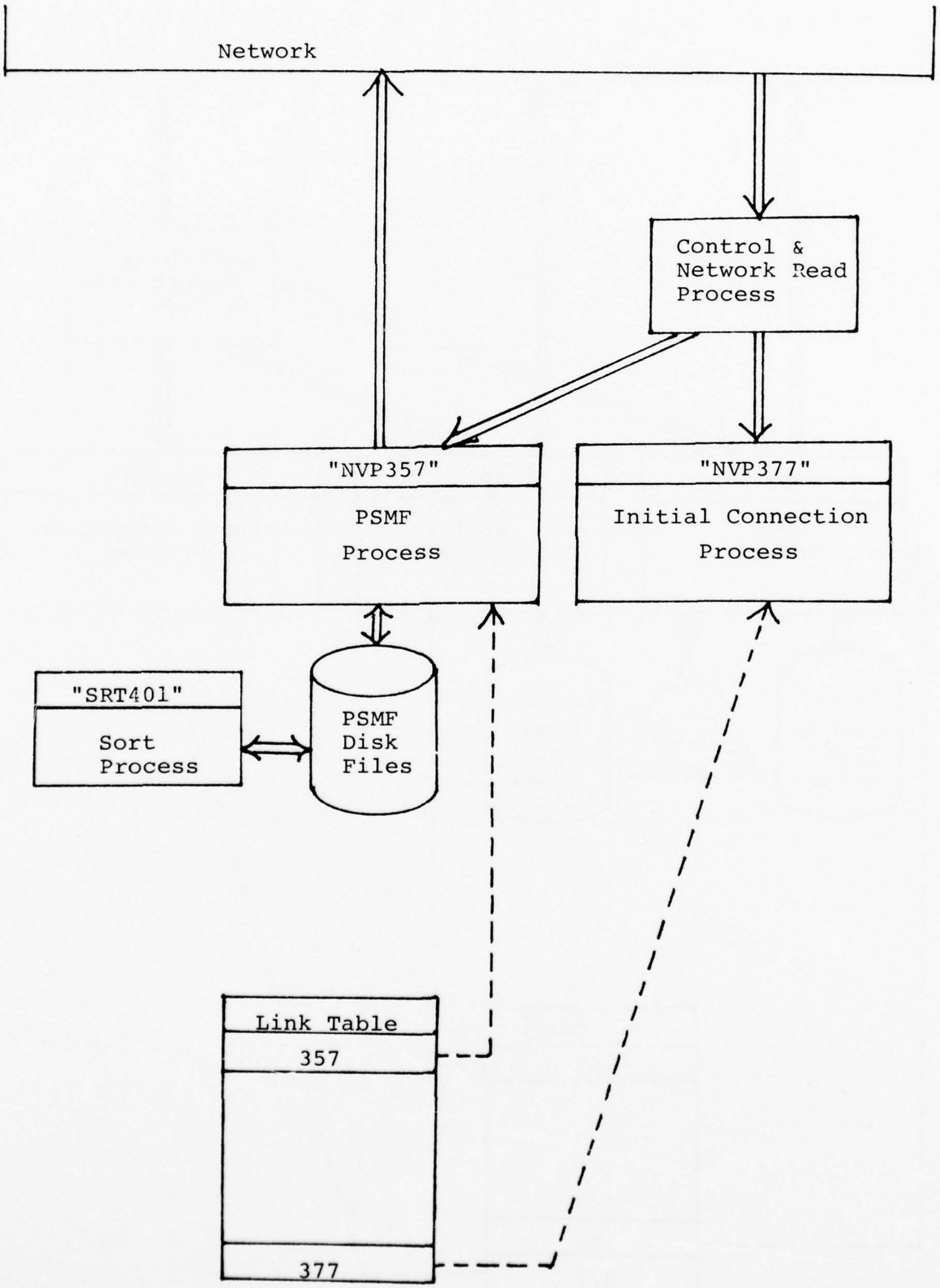
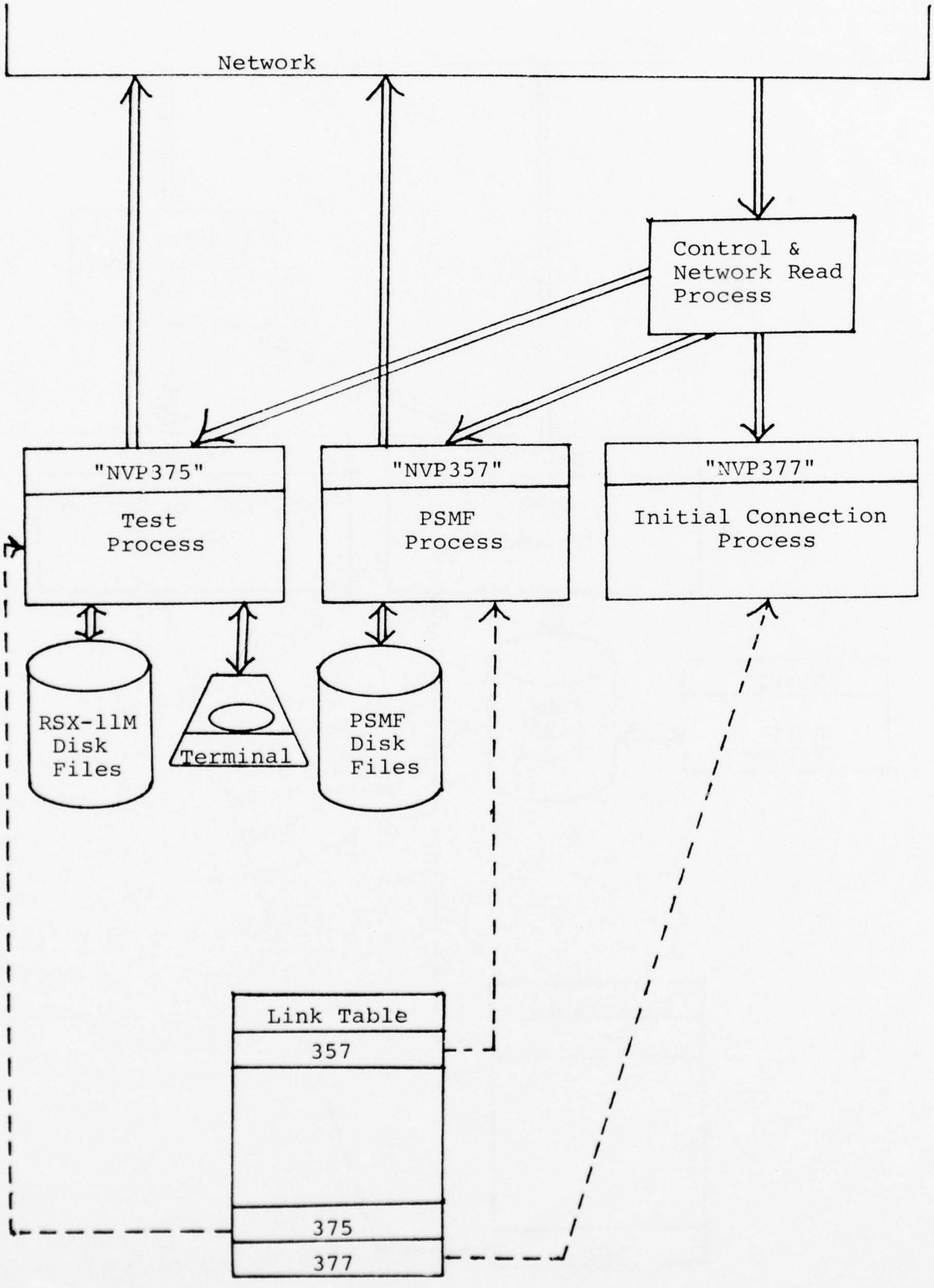

Network
Control &
Network Read
Process
"NVP357"
PSMF
Process
"NVP377"
Initial Connection
Process
"SRT401"
Sort
Process
PSMF
Disk
Files
Link Table
357
377

Figure 3.6


Network
Control &
Network Read
Process
"NVP375"
Test
Process
"NVP357"
PSMF
Process
"NVP377"
Initial Connection
Process
RSX-11M
Disk
Files
Terminal
PSMF
Disk
Files
Link Table
357
375
377

4. Implementation Specifics

This section describes in some detail the current level of PSMF implementation. Section 4.1 discusses the data structure design for PSMF processes. Section . 4.2 describes the PSMF file structure. From there, a description of constituent modules ensues.

4.1 PSMF Processes Data Structures

It was envisioned from the first that the PSMF would eventually support multi-user activity. This anticipation led to the design of a coherent data structure for all processes.From the standpoint of inter-process communication, it was desirable to have these data structures in the address space of all processes. These considerations led to the following implementation:

a. All processes have associated entries in a "Link Table". For processes which are actually involved in network communications, Link Table positions map into the listening NVP links for that process. For processes which do not need to communicate, pseudo link positions greater than 400 are allocated.

b. A process's Link Table entry points to its "Usertable", which contains the fundamental data structure for the process. The Usertable is allocated from the Common buffer area when the process is initiated, and de-allocated when the process terminates.

c. All other data structure requirements for a process (e.g. disk buffers) are met by allocations from the Common buffer area.* Pointers to these areas can ultimately be found in the Usertable.

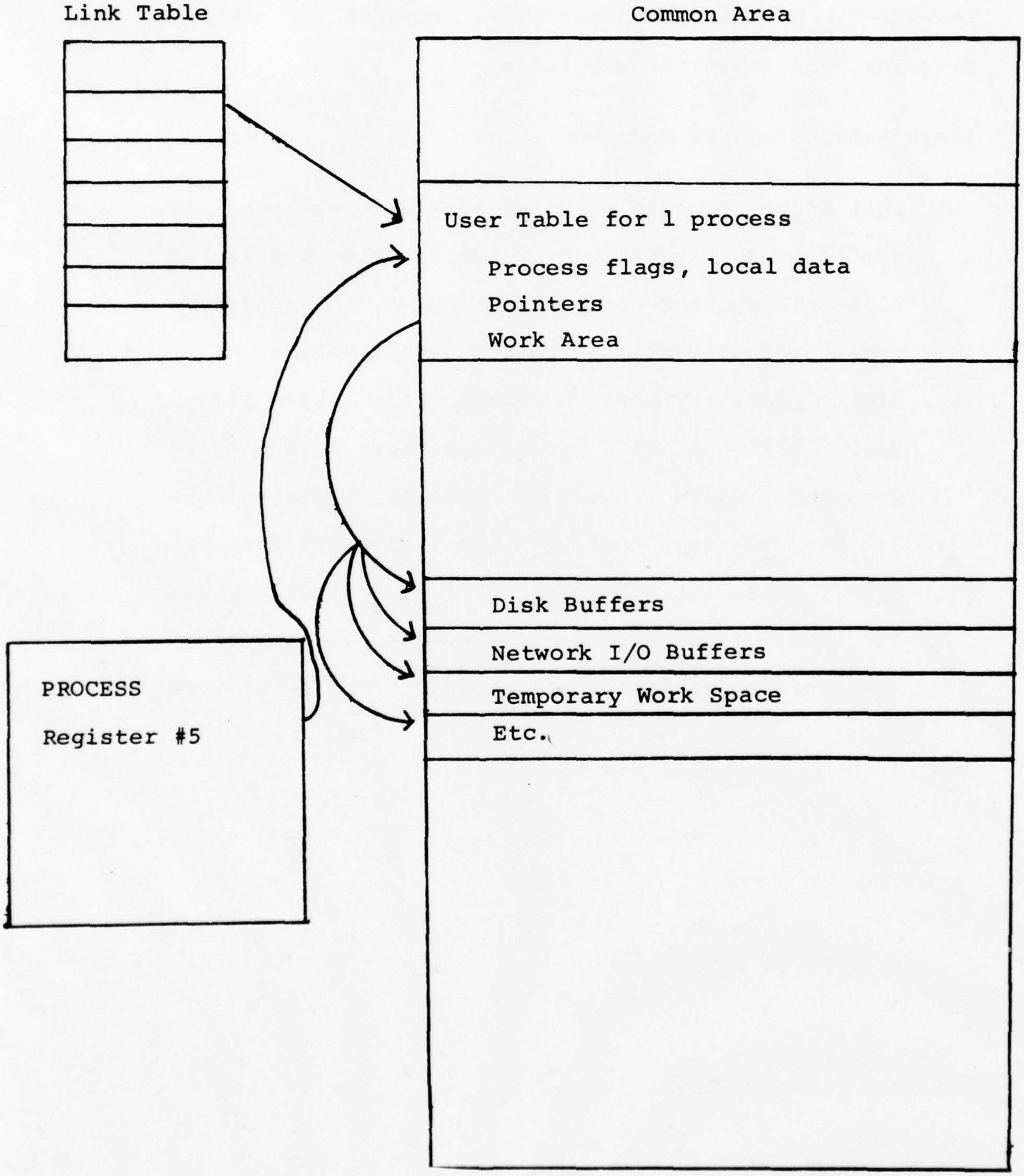
d. Convention dictates that a process access its Usertable using indices relative to a single dedicated register. This convention facilitates intra-process communications.

Figure 4.1 illustrates these data structures.

With the exception of the Test process, and parsing facility which have unconventional needs for RSX associated structures.

Figure 4.1

PSMF Process Data Structure



In addition to the data represented in each Usertable, there is a need for global data which can act as semaphores. This need is met by the RSX "event flags" provision, which includes executive services for setting, clearing, and awaiting such flags.

These semaphores are used to:

a. Lock access to data structures to prevent possibly conflicting alterations. For example, the linked list of buffers in Common have a semaphore associated with allocation and de-allocation.

b. Signal processes that an event has taken place. Each of the PSMF processes has a dedicated semaphore, which is used to indicate that network I/O has been completed, or that a time delay is up.

c. Signal a request for services. The Control process can, for instance, be awakened by its dedicated semaphore to communicate a request for a network read.

4.2 PSMF File Structure

Data pertinent to PSMF storage, retrieval, and measurement is of two fundamentally distinct types. Firstly, there is the actual stream transmitted by the user. Secondly, there is information related to arrival time and sequence. This dichotomy encouraged the following implementation:

a. Any recording session results in two PSMF files: a "primary" file and a "secondary" file.

b. The primary file consists simply of the stream of data transmitted by the user. Any network responses generated by the PSMF during negotiation are also recorded - to serve both as a debugging record and a guide during playback.

c. The secondary file contains the recording date and time, user identification, parcel timing, and file size. In addition, for every message in the primary file, the secondary file contains various flags, PSMF time stamp, and sorted link to next (by user's time stamp) message.

It is intended that the secondary file contain all information needed by the measurement functions, as well as that necessary for ordered playback.

4.3 Network Driver

The network driver is the routine in the RSX-11M executive which provides full duplex access to the Arpanet. This access is achieved via standard RSX "QIO" executive requests. RSX provides the mechanism for queueing requests and communicating parameters to the driver. It is the driver's responsibility, on the other hand, to signal normal or abnormal request completion.

The PSMF driver was designed to treat the interface as two separate devices - one each for input and output. Completion of a type 3 network write (uncontrolled) can be signalled merely by completion of physical transfer to the IMP. Completion of a type 0 network write, however, requires reading a corresponding RFNM from the destination IMP. This means that, in general, processes do not interface directly to the network driver, but are filtered through Library routines which cooperate with the Control process.

4.4 Library Routines

These routines consist of re-entrant and position independent code which is shared by all PSMF processes. While such an entity is not strictly essential, it does serve to reduce core requirements and compartmentalize the use of requests for executive services.

There are many Library routines. They are briefly described below.

4.4.1 Network Read

This routine will suspend a process if a previous network read has not been completed. It then signals the Control process that data is expected from the network.

4.4.2 Network Write

Type 3 writes and Type 0 writes are treated independently. The Library routines will wait on a write of a given type if a previous write of that same type has not been completed. A QIO is then issued to start the write, and, in the case of a write type 0, the Control process is notified that a RFNM is expected.

In addition, Library routines are available to move data to a temporary buffer before starting the write. This avoids locking the process's own buffers while awaiting completion.

4.4.3 Wait on Network I/O

These routines are called by a PSMF process or by other Library routines to wait until completion of a network read, write type 0, or write type 3. If no I/O is in abeyance, no wait is effected.

4.4.4 Usertable & Process Initiation

If a process is initiated by another process, some communication of data is usually necessary. A PSMF initiating process will allocate and set up a Usertable for the initiated process, start that process, and communicate the Usertable address. Library routines perform all these functions.

4.4.5 Time Interval Computation

A process's initiation time is stored in its Usertable. A Library routine is available to compute, in milliseconds, the difference between the time of call and the time of initiation. This is used as the PSMF time stamp during the PSMF Record function.

4.4.6 Timed Suspension of a Process

This routine will suspend a process for a specified number of milliseconds. It is used by the PSMF Playback function to time the transmission of encoded voice packets.

4.4.7 Management of Common Buffer Pool

The common pool of fixed length buffers supplies Usertables, network I/O buffers, disk buffers, and temporary work space. A pair of Library routines are used to allocate and free such buffers.

4.4.8 Process Exit

This routine frees all the buffers owned by a process, resets its link table entry, and performs an RSX exit call.

4.5 Control Process

The Control process is initiated at an RSX-11M terminal, or by the system clock queue. Its functions are threefold: it initializes PSMF resources, acts as a network read steerer, and adjudicates network I/O completion.

4.5.1 Initialization of PSMF Resources

As the first PSMF process to be invoked, it is the Control process's job to initialize resources:

a. Interprocess semaphores are reset.

b. The Link table is cleared.

c. A Common buffer region is dynamically created and a mechanism is set up to allow all future processes to map to it.

d. The IMP is primed with a few NOPS.

e. The Initial Connection Listening Process is initiated.

f. A network read is started.

4.5.2 Network Read Steerer

The initial network read will wake up the Control process when two words (32 bits) have been transferred into a special header buffer. The Control process analyzes this header to effect the disposition of the rest of the message:

a. Unrecognizable headers cause the remaining message to be discarded.

b. Links on type 0 and 3 messages are examined.

    i) If it is not an NVP link, the message is discarded.

    ii) If the link is not owned by a PSMF process, the message is discarded.

    iii) If the owning PSMF process had requested a network read, the remaining message is read into the area designated by the process.

    iv) Otherwise a free buffer is allocated, the remaining message is read into it, and the process is flagged as having received an unsolicited read. This facility is designed to allow the Recording function to fall a little behind in its retrieval of type 3 messages.

c. A NOP is ignored.

d. A RFNM is used to flag the corresponding network write type 0 as complete.

e. An "incomplete transmission" is ignored. Failure of a type 0 write is noted when no RFNM arrives within five seconds.

f. An "interface reset" forces transmission of a few NOP's and discarding the first input.

g. All other messages are ignored.

4.5.3 Adjudication of Network I/O Completion

The Control process is notified of all requests for network reads.

a. If an unsolicited read has already been undertaken on behalf of the requesting process, its data are moved to the requested area and the unsolicited buffer flushed.

b. Otherwise the next network read with an appropriate link is steered into the requested area.

In addition, the Control process is notified of all completed network I/O. This provides a central point for resolution of PSMF process waits on network I/O. For example, if a process requests a wait until completion of network write type 0:

a. The request is made through a Library routine which checks if such a write has been requested but not finished. If not, an immediate return is effected.

b. The Library routine then clears a global semaphore assigned to the PSMF process, sets a "wait" flag in the process's Usertable, and issues an RSX executive directive. RSX will resume the process when the global semaphore has been set.

c. When the Network Read Steerer section of the Control process detects the appropriate RFNM, it sees the PSMF process's wait flag, and sets the proper global semaphore. RSX is thus signalled to awaken the PSMF process.

4.6 Initial Connection Network Listener

NVP initial connection protocol requires an opening request on link 377. Since PSMF processes are in general associated with links, it was convenient to assign the initial connection function to a distinct process. This Listener process is initiated by the Control process on PSMF initialization (3.5.1) and performs the following duties:

a. It requests a network read on its link of 377 and waits pending read completion.

b. When a message with the correct format is received, the Link table is searched for the first free entries. Such entries are mapped to a link number, e.g. link 357.

c. A process associated with that link number is initiated, depending on the nature of the message, e.g. "NVP357", "VCP357" OR "ECH357" for PSMF, NVCP record, or non-PSMF echo respectively.

d. The Listener process allocates and sets up a Usertable for the new process, sends the Usertable address, and goes back to the listening start. It is the new process's responsibility to carry on all communications on its own links.

4.7 PSMF File Accessing Routines

These are a series of modules designed to interface between a PSMF process and RSX-11M file-handling primitives. While suitable for Library inclusion, space limitations now dictate that these routines be linked separately with each file handling PSMF process.

Each routine deals with a single file at a time, e.g. with either a primary or secondary file. Access is by record at the PSMF level and by block at the RSX level. Several processes can simultaneously access a single file.

An outline of the routines follows.

4.7.1 File Open Routines

A PSMF file may be opened for writing (record), reading (playback, etc.), or updating (append). In each case, the using process supplies the record size, file name, and the file number (0 thru 3) which will be used for all future accesses.

A file open routine will also allocate Common buffer space for RSX specific file information, and two block buffers for each file.

4.7.2 File Close Routine

The file close routine will initiate an RSX file close and free buffer space used by the file.

4.7.3 Disk Read Routines

The disk read routines require a file number and record number for access. A pointer to the requested record is returned. As an option, the sequential read routine tries to insure that a record's predecessor and successor are in core in anticipation of future reference.

4.7.4 Disk Write Routines

The PSMF disk write routines write only out of the "current" record buffer; in order to change a record, that record must first be read. As an option, the sequential write routine increments the "current" record number and buffer address after each write.

Both the read and write routines employ two block buffers (16 primary file records or 64 secondary file records). No physical disk transfer take place if either of the two block buffers contains the desired record.

4.8 Record Function

The PSMF Function process (Record and Playback) is initiated by the Network Listener process (3.6). The Function process continues the opening protocol and awaits a PSMF command.

The PSMF Record command has provision for a filename and optional password. The Record function is called and:

a. Opens two PSMF files: one with 64-word records and the extension ".PR" (primary file) and the other with 8-word records and the extension ".SE" (secondary file).

b. Sends an acknowledgement to the user.

c. Records the PSMF Record command as the primary file's first record.

d. Records the date, time, and user ID as the first record of the secondary file. Parcel timing and file size information is added to this record at the conclusion of the recording session.

The Record function then goes into a network listen loop in which certain messages are recorded on the disk. These messages which are stored are:

a. PSMF "field to be recorded" messages.

b. All acceptable negotiations for voice transfer. The Record function will accept all suggestions (except for message size greater than 976 bits.)

c. All voice negotiation replies from the PSMF to the user. These are flagged appropriately and are used as a guide during Playback.

d. When sufficient information for Playback timing is available, the user can terminate negotiation and record voice messages.

Every stored message has an entry in each of the primary and secondary files.

a. The entire message is stored as a single record in the primary file as it is received, preceded by a word containing its length.

b. The corresponding record in the secondary file contains a field name word, a flag word, the PSMF time stamp in milliseconds, words 3 and 4 of the message, and provision for a link to the next record (to be sorted by user's time stamp by the "Sort by Time Stamp" process).

The Record function remains active until receipt of an NVP "goodbye" or PSMF "end of file" message. In either case, the primary and secondary files are completed and closed. The Sort process is initiated in order to link the secondary file by user's time stamp. A "goodbye" will terminate the NVP connection, while an "end of file" will cause control to return to the start of the PSMF Function process.

4.9 Sort Process

The Arpanet does not guarantee the integrity of a stream of type 3 messages. Voice packets may arrive at the PSMF out of sequence, as duplicates, or may not arrive at all.

It is the function of the Sort process to link a PSMF message file so that Playback message sequence will closely approximate that in which they were sent. This is accomplished by

a. Linking the voice messages by the sender's time stamp.

b. Linking imbedded PSMF "field to be recorded" messages as propitiously as possible.

The Sort process is initiated by the record function at the conclusion of a recording session. The Sort process has no requirements for network communication, so is given a link table entry corresponding to a network link greater than 400 (see section 3.1). It opens the secondary file with shared access (to allow concurrent playback) and performs the following functions:

a. An expected time stamp is computed from the present time stamp and parcel count.

b. A forward search of the secondary file is made up to a one minute window boundary. If a control message (imbedded or preliminary) is encountered immediately, it is linked and the entire search begins again. If the expected time stamp is encountered, it is linked and a new search begins.

c. Otherwise a search in the negative direction up to a one-minute window boundary is made. If the expected time stamp is encountered, it is linked and a new search begins.

d. Otherwise a link is made to the record with the time stamp differing least from the expected time stamp, if any.

e. Otherwise the sort terminates. (No more records).

This algorithm permits linkage of files consisting entirely of control messages as well as those with duplicate or missing messages.

Sorting of a file produced by the NVCP recording function proceeds in an analogous manner. A "61,6" PSMF end of file message terminates each speaker's segment in this case, and the Sort process considers this message to be the last record of each segment.

4.10 Playback Function

Once a PSMF file has been created by the Record function and linked by the Sort process, it can be accessed by the Playback function of the PSMF Function process.

The PSMF Playback command has provisions for filename and password. When this command is encountered by the PSMF Function process,

a. An attempt is made to open the primary and secondary files associated with the filename. If this attempt is unsuccessful, a negative acknowledgement is sent to the user.

b. The first record of the primary file is read to determine the password. If no password was used for recording, no check is made of that included with the playback command.

c. A password comparison is made. If a bad match is found, a negative acknowledgement is sent to the user.

d. Else a positive acknowledgement is sent and the Playback function awaits user commands specifying the information to be retrieved.

The user specifies a list of his selections in a PSMF "fields to be retrieved" command. The Playback function then performs the following:

a. If the user has specified voice, the PSMF conducts negotiations as per those which were recorded. If the user disagrees with any of the recorded negotiations the playback session is terminated.

b. The secondary file is scanned from the beginning to find any message types on the user's list. If found, the corresponding record in the primary file is transmitted.

c. The interval timing between voice message transmission is computed from the parcel count and parcel timing data.

d. If no further messages are to be retrieved, or if the user sends a PSMF "end of file" command, Playback sends a count of messages transmitted and awaits a further list of selections from the same file. An NVP "goodbye" elicits the same count and a termination of the Playback session.

4.11 Append Function

The PSMF Append function will permit data to be added to an existing file and will alter that file to reflect the addition.

a. The first record of the secondary file is flagged to indicate an appendage.

b. The original file is scanned for any message which were flagged as requiring a reply (negotiations). These are sent as negotiations to the user, and the appropriate response (previously recorded as "our response") is expected.

c. At the completion of negotiations, the file is positioned to its end, and recording proceeds using the same control structures as the PSMF Record function.

4.12 Delete Function

The PSMF Delete function can be used to delete PSMF files.

a. The password is checked.

b. The requesting Host/Imp/Extension is compared with the creating Host/Imp/Extension.

c. If everything is okay, both the primary and secondary files are deleted.

4.13 Measurement Function

As described in the previous sections, the Record function creates a PSMF file which can be accessed via the Playback function. In an analogous fashion, the Measurement functions take extant files as input and create PSMF files as output. These output files can be retrieved by Playback, or can act as input to further Measurement functions.

The PSMF "Open input for measurement" command has provisions for filename and password. When this command is received by the PSMF function process,

a. An attempt is made to open the primary and secondary files associated with the filename. The first record of the primary file is read and the password checked. If unsuccessful, a negative acknowledgement is sent to the user. Else the primary file is closed (all pertinent measurement data is stored in the secondary file), and an acknowledgement sent to the user.

b. The PSMF Function process awaits receipt of an "Open output measurements" command. This command has provisions for output filename and password, as well as information concerning the measurement to be performed.

c. The output primary and secondary files are opened, an acknowledgement is sent to the user, and the files are initialized as in the Record function.

d. The first word after the password is used to denote the <measurement-specification>. Following double-word <measurement-modifiers> may contain data for the particular measurement function.

AD-A052 338   COMPUTER CORP OF AMERICA CAMBRIDGE MASS   F/G 17/2
A PACKET SPEECH MEASUREMENT FACILITY: FINAL TECHNICAL REPORT, 1--ETC(U)
FEB 78 D A LOW   MDA903-77-C-0156
UNCLASSIFIED   CCA-78-05   NL

2 OF 2
AD
A052338

END
DATE
FILMED
5-78
DDC

4.13.1 Relative Delays

If the <measurement-specification> is 0, the measurement of relative network delays is invoked.

a. The input secondary file is scanned in sorted order.

b. For each voice packet encountered, a transformation is made on the user's time stamp into a two-word integer.

c. For each voice packet, a relative delay is computed as

(This PSMF Time Stamp
-
First PSMF Time Stamp) -
(This User's Time Stamp
-
First User's Time Stamp)

d. The relative delay is stored in the primary and secondary output files as a "61,5,3" (binary) message with a doubleword predicate. The secondary file also contains the original PSMF time stamp, for re-construction of "time into message".

e. When the end of the input file is reached, all files are closed and the PSMF awaits a function specification.

4.13.2 Missing, Duplicate, or Out of Order

If the <measurement-specification> is 1, the measurement of missing, duplicate, or out of order is invoked.

a. The input secondary file is scanned in received (unsorted) order.

b. If a voice packet was flagged as requiring a backward link by the Sort process, it generates an "out-of-order" (1) output message.

c. If a voice packet was flagged when the Sort process could not find the next expected time stamp, and the next actual voice packet has no "we-skipped-parcels" bit, it generates a "missing" (3) output message.

d. If a packet was not linked by the Sort process, it generates a "duplicate" (2) output message.

e. All other packets generate an "other" (0) output message.

f. Generated messages (one or two per input packet) are assembled into "61,5,3" messages with doubleword predicate, and written to the output files. The secondary output file also contains the PSMF time stamp for reconstruction of the "time into message".

g. When the end of the input file is encountered, all files are closed and the PSMF awaits a function specification.

4.13.3 Histogram

If the <measurement-specification> is 2, a histogram is computed. In this case, two doubleword <measurement-modifier>s are assumed to follow the <measurement-specification>. The first is taken to be the lower histogram boundary, and the second to be the interval size. One hundred and twenty eight "bins" are created from this information. A buffer from the Common Area is used to contain the bins.

a. The input secondary file, which is typically the output file of another measurement function, is scanned in unsorted order.

b. The doubleword value (which is equivalent to the "61,5,3" predicate in the primary file) is used to increment the corresponding histogram bin.

c. When the end of the input file is encountered, the 128 histogram bins are written as predicates to "61,5,3" messages on the output files. The secondary output file also contains lower bin boundaries. All files are closed, and the PSMF awaits a function specification.

4.13.4 Periods of Speech and Silence

If the <measurement-specification> is 3, a computation is made of periods of speech and silence. This function is dependent on the "we-skipped-parcels" bit in voice packets, and is actually a measure of this parameter.

a. The input secondary file is scanned in sorted order.

b. When a voice packet is encountered with a "we-skipped-parcels" bit on, two output messages are created. The first, which has a <field-select> = 1, is the duration (in milliseconds) of the "speech" period up to, but no including, this voice packet. The second output measurement, which has a <field-select> = 2, is the duration of the "silence" period represented by this packet.

c. When the end of the input file is encountered, a final "speech" duration measurement is made, all files are closed, and the PSMF awaits a function specification.

4.13.5 Field Type Classification

If the <measurement-specification> is 4, all input packets are classified into field types. The output files consist of messages containing the corresponding classifications.

a. The input secondary file is scanned in unsorted order.

b. Each input packet is classified into: 0 - voice; 2...4 - PSMF control of form "61,5,(2...4)". Output messages contain this numerical classification as predicates of "61,5,3" messages.

c. When the end of the input file is encountered, all files are closed and the PSMF awaits a function specification.

4.13.6 Mean and Standard Deviation

If the <measurement-specification> is 5, input "binary" ("61,5,3") messages will be subjected to an analysis yielding the mean and standard deviation of their predicates. A modified DEC floating point software package is used for these computations.

a. The input secondary file is scanned in unsorted order.

b. Each binary control message is counted, and its predicate converted to single precision (standard PDP-11) floating point format. A sum and sum of squares is updated.

c. At the end of the input file, three output messages are created, in which the binary predicate is a number in floating point representation: the count, mean, and standard deviation of the input data. All files are closed and the PSMF awaits a function specification.

4.14 NVCP Recording

The NVCP Recording function was not implemented in the PSMF function process. The reasons for this were as much aesthetic as practical: the specification for NVCP recording filename is in the invitation to join the conference, rather than after completion of the initial connection procedure. In addition, adherence to NVCP rules forces the PSMF to act as a caller, rather than answerer, after the initial invitation.

The NVCP recording function was thus implemented as a separate process from the PSMF function process. It is requested by the Network Listener (Section 4.6) on receipt of a [42] "Please join my conference" message which has a <filename-string>, <password-string> appended.

a. The NVCP recording function then calls the chairman on link 377 to initiate a connection.

b. Voice negotiations are carried on, with the chairman as negotiation master. The NVCP recorder copies these negotiations to the PSMF file exactly as in a standard PSMF recording session.

c. When negotiations are complete, the chairman sends a [34] "Add a participant to my conference" message to the PSMF. This is converted to a [8] "Ready?" before recording (for compatibility with the PSMF playback function).

d. The NVCP recorder signifies its readiness and starts to record voice messages.

e. Every time the chairman sends a [36] "Listen to ... (stream #0)", the NVCP closes the previous section of the file with a "61,6" (EOF) message and appends a new section, exactly as in the PSMF Append function.

f. On receipt of a "2" (Goodbye) message, the file is closed and the standard Sort process is started. The resulting sorted file is amenable to PSMF playback and measurements.

4.15 Non-PSMF Echo

The Non-PSMF echo function does not employ any of the PSMF file utilities and was thus implemented as a separate process from the PSMF Function process.

The non-PSMF echo process is initiated when the network listener process receives a "1" (calling) message where the <whom> has extension 0. Standard NVP is used.

4.16 Parsing Facility

In order to facilitate PSMF usage, a parsing facility was implemented to create PSMF function commands out of a PSMF <string>. This facility is invoked when a "61,8,<string>" message is received while awaiting a PSMF function specification. Details on use of this facility can be found in Section 2.4.

Implementation of this facility was accomplished almost entirely through the use of a pre-packaged RSX-11M parser. After a successful parsing attempt has been used to create a standard PSMF function message, that message is sent by the parser to the PSMF via the IMP interface. Positive or negative acknowledgements to these constructed messages are returned to the user.

4.17 Test Process

The Test process was designed to provide a full testing facility for the PSMF implementation. It essentially consists of highly RSX dependent code, initiated from an RSX terminal, which can read or write RSX files and access the Arpanet as a PSMF process.

When initiated, the Test process:

a. Takes NVP links 375 and 376 for its own read links.

b. Asks for an input RSX file to read data and an output file for results.

The input data can be commands to:

a. Affect network output: i.e. change destination HOST/IMP, output link or message type.

b. Read network input.

In addition, data to be transmitted over the network can consist of strings

N(1), N(2),...,N(K)

where each N(I) is the octal representation of a PDP-11 word or byte, or an ASCII string enclosed in double quotes.

These data strings are converted to an Arpanet message and transmitted, typically through the IMP and back to a PSMF process. Data received from a network read are converted to similar readable strings and sent to the Test process output file. This permits interactive (via a terminal) or high speed (via an RSX disk file) testing of PSMF constituents.

5. Future Work

The facilities described in the previous sections are currently available through the Network Voice Protocol and its PSMF extension. This protocol, unfortunately, imposes considerable restrictions on the nature of the information which can be communicated. In particular, the PSMF is concerned with file handling and a variety of measurement procedures. These procedures would be more accessible if invoked through the auspices of a Network Control Program and an associated TELNET and FTP. CCA will install such facilities on the PSMF machine.

Access to the measurement facility through TELNET will take place via a sophisticated high-level language extension. This extension will be a flexible one and will be applicable to experiments recorded under a variety of protocols.

The installation of standard ARPANET host-host will facilitate PSMF communication with the CCA DATACOMPUTER. This will permit storage and retrieval of an extremely large experimental database, as well as enabling an automatic backup and archival facility.

As in the first year, it is anticipated that PSMF development will be responsive to user needs. This approach is consonant with the intent to make the PSMF a valuable tool to network researchers of all disciplines.

Appendices

The purpose of these Appendices is to present some examples of the measurement functions currently available to PSMF users, and to demonstrate the kinds of questions the measurements can initiate as well as answer.

Appendix A deals with a series of experiments conducted by researchers working with Randy Cole at ISI during December 1977 and January 1978. The purpose of this series was to detect any correlation between network induced perturbations and time-of-day or day-of-week.

Appendix B is concerned with experiments conducted by the same group during January 1978, which were designed to investigate network perturbations as a function of packet size/frequency.

A.

During December 1977 and January 1978, a series of thirty-one files were recorded by ISI researchers located in Marina del Rey, California, at the PSMF located at Cambridge, Massachusetts. Each file represented a monologue of approximately 5 or 6 minutes length and consisted of about 1000 packets of 1000 bits sent 2.7 packets per second. Voice packets were sent as Arpanet type 3 (uncontrolled) messages.

Files were recorded on Mondays through Fridays at three times: "AFT"=14:00 PST, "EVE"=17:00 PST and "NIT"=19:00 PST. Figure A1 shows the matrix of recorded files.

Figure A2 is a table of measurements obtained using standard PSMF functions. The rows are associated with individual files. The columns represent, left to right:

1. The count of voice (type 3) packets in the file.

2. The mean relative delay of all type 3 packets (in milliseconds) in the file. Since no universal clock is available, absolute network delay cannot

Experiment Files Figure A.1

EXPERIMENT FILES

	AFT	EVE	NIT
M	JAN168 JAN238	DEC197 JAN168	DEC197 JAN168
T	DEC207 DEC277 JAN108	DEC207 DEC277 JAN108	DEC207 JAN108 JAN248
W	DEC217 DEC287	DEC287	DEC217
T	DEC297 JAN058	DEC297 JAN058 JAN268	JAN058 JAN268
F	DEC237 DEC307 JAN068	JAN068	JAN068

SUMMARY OF DAILY EXPERIMENTS

DEC 19, 1977 - JAN 26, 1978

FILE	COUNT	MEAN R.DELAY	S.D. R.DELAY	RANGE	MISSING	DUPLI-CATE	OUT OF ORDER
DEC197EVE	999	-74	131	970	0	0	0
DEC197NIT	985	88	151	1309	1		8
DEC207AFT	1022	-122	125	1094	3		2
DEC207EVE	999	-96	126	818	0		1
DEC207NIT	996	7	118	820	1		1
DEC217AFT	996	-19	123	847	3		2
DEC217NIT	998	-157	173	1328	0		6
DEC237AFT	1406	-4	123	855	2		2
DEC277AFT	1005	-77	137	1033	1		1
DEC277EVE	1099	-3	116	817	3		2
DEC287AFT	971	-225	135	1181	0		4
DEC287EVE	994	-70	121	1133	1		1
DEC297AFT	995	-121	123	922	3		0
DEC297EVE	1044	-15	112	692	1		0
DEC307AFT	992	-146	124	1060	1		0
JAN058AFT	999	-53	110	810	0		1
JAN058EVE	997	19	102	906	2		0
JAN058NIT	1004	-90	118	912	0		0
JAN068AFT	1020	41	125	1078	0		0
JAN068EVE	993	-110	122	843	1		2
JAN068NIT	1025	-27	95	671	1		1
JAN108AFT	1056	-71	103	786	2		0
JAN108EVE	1041	36	92	636	1		0
JAN108NIT	1016	11	95	652	1		0
JAN168AFT	1054	-32	103	772	3		0
JAN168EVE	1001	-155	97	888	4		0
JAN168NIT	978	51	93	735	2		0
JAN238AFT	1016	112	152	1102	1		2
JAN248NIT	973	0	96	695	2		0
JAN268EVE	998	-148	115	996	2		0
JAN268NIT	985	34	94	648	4		0
	1021	-46	118	893	1.5		1.2

be determined. Packet delays "relative" to that experienced by the first packet:

relative delay =
(received time packet N -
received time packet 1)
- (sent time packet N -
sent time packet 1)

are the source of all delay data. The mean of such data is fairly useless, except as a way of indicating a bias. Such a bias might be seen here, with most voice packets experiencing less network delay (negative relative delay) than the first.

3. The standard deviation of relative delay (in milliseconds). Since standard deviation is invariant with respect to an additive constant, this is also the standard deviation of absolute network delay.

The data show that, if a normal distribution is assumed, a time slot of 240 milliseconds (± 1 standard deviation) will include 70% of arriving packets.

Figure A3 shows the standard deviations arranged into the time matrix. A Bartlett's test for

Standard Deviation of Relative Delays Figure A.3

RELATIVE DELAYS
STANDARD DEVIATION

	AFT	EVE	NIT
M	103	131	151
	152	97	93
T	125	126	118
	137	116	95
	103	92	96
W	123	121	173
	135		
T	123	112	118
	110	102	94
		115	
F	123	122	95
	124		
	125		

HYPOTHESIS THAT VARIANCES ARE EQUAL CAN BE REJECTED (BARTLETT'S TEST) homogeneity of the associated variances was run and showed that the hypothesis of homogeneity could be discarded at a very high level of confidence.

4. The range of relative delay. Again, this is the same as the range of absolute delay. The data show a range of about a second for each file of 1000 packets. An analysis of variance on the time matrix of Figure A4 showed no correlation of this parameter with time of day or day of week.

5. The number of missing packets. This was determined from the time stamps (which are also sequence counts) of the packet stream. The data show about 1.5 missing packets for each 1000 packet file.

6. The number of duplicate packets. No duplicates were found in any of the experiments.

7. The number of out of order packets. This statistic was also determined from the transmit time stamp. The data show about 1.2 out of order packets for each 1000 packet file. Figure A5 represents the time matrix of missing and out-of-order packets.

Range of Relative Delay Figure A.4

Range of Relative Delay

	AFT	EVE	NIT	$\bar{X}$	σ
M	772 1102	970 888	1309 735	963	215
T	1094 1033 786	818 817 636	820 652 695	816	157
W	847 1181	1133	1328	1122	201
T	922 810	692 906 996	912 648	840	129
F	855 1060 1078	843	671	901	170
X	961 144	870 145	863 272	893	175

Out of Order Messages Figure A.5

MISSING - OUT OF ORDER

	AFT	EVE	NIT
M	3-0	0-0	1-8
	1-2	4-0	2-0
T	3-2	0-1	1-1
	1-1	3-2	1-0
	2-0	1-0	2-0
W	3-2	1-1	0-6
	0-4		
T	3-0	1-0	0-0
	0-1	2-0	4-0
		2-0	
F	2-2	1-2	1-1
	1-0		
	0-0		

$\bar{X}$ = 1.5 - 1.2
σ = 1.2 - 1.9

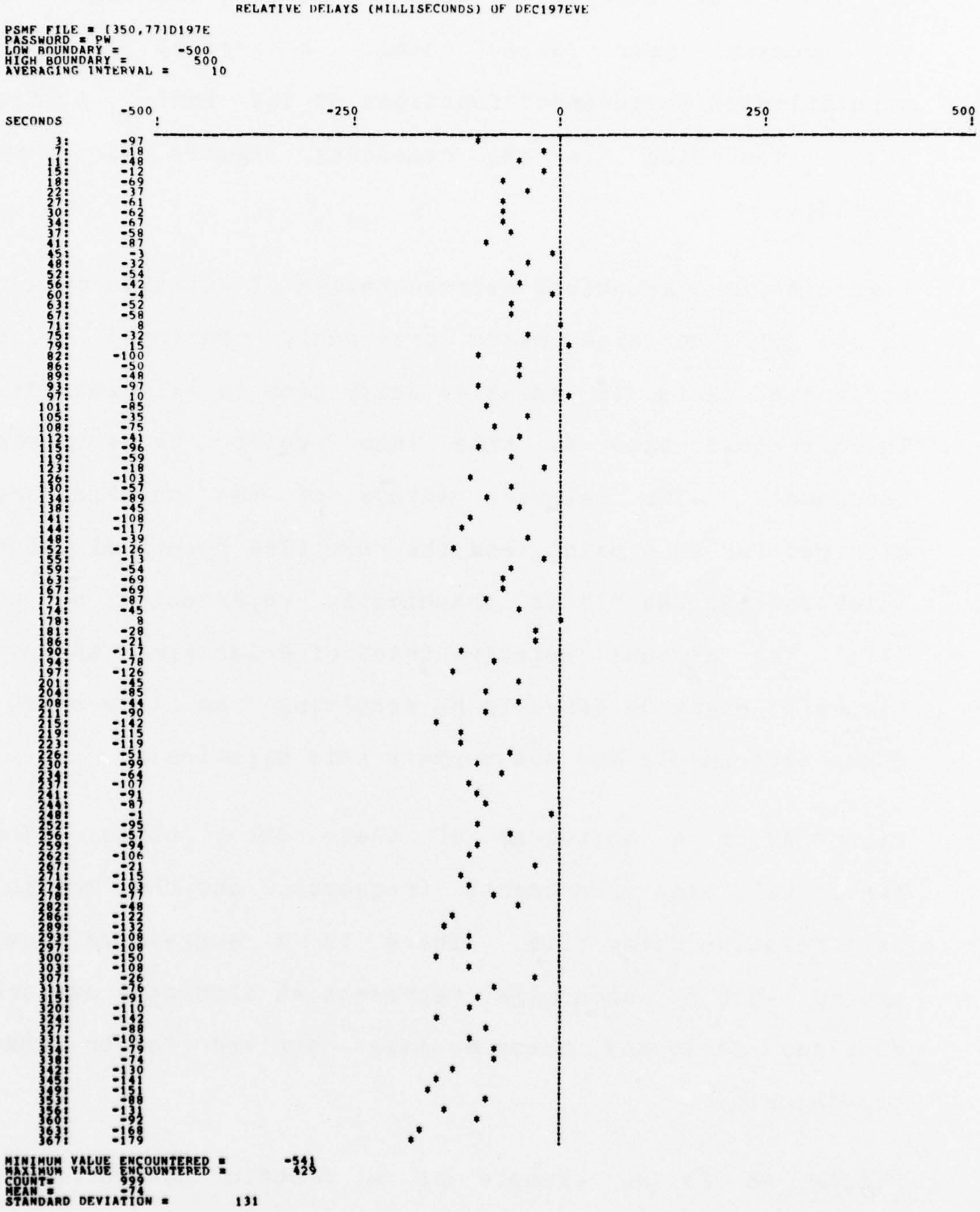
The planned availability of standard network software in the coming year will enable a variety of more sophisticated measurement functions on the PSMF. A few are illustrated in the remaining figures for this appendix.

Figure A6 is a graphical representation of relative delays in one of the experiments previously mentioned. The horizontal axis is relative delay time in milliseconds. The vertical axis is time into voice transmission (seconds). The relative delays of ten packets are averaged for each point, and the resulting numerical value which follows the ":" is graphically represented by an "*". The gradual negative trend of delay times seen in Figure A6 might be taken to be resulting from clock shift. Other experiments did not support this hypothesis.

Figure A7 is a histogram of these delay times. The horizontal axis represents frequency, and the vertical axis relative delay time. There is a suggestive lump around -400 ms which might represent an alternate network routing. Curiously, these messages arrived faster than the majority.

Figure A8 is an example of a function not currently available under the PSMF extension to NVP. It represents the correlation coefficients of packet relative delay

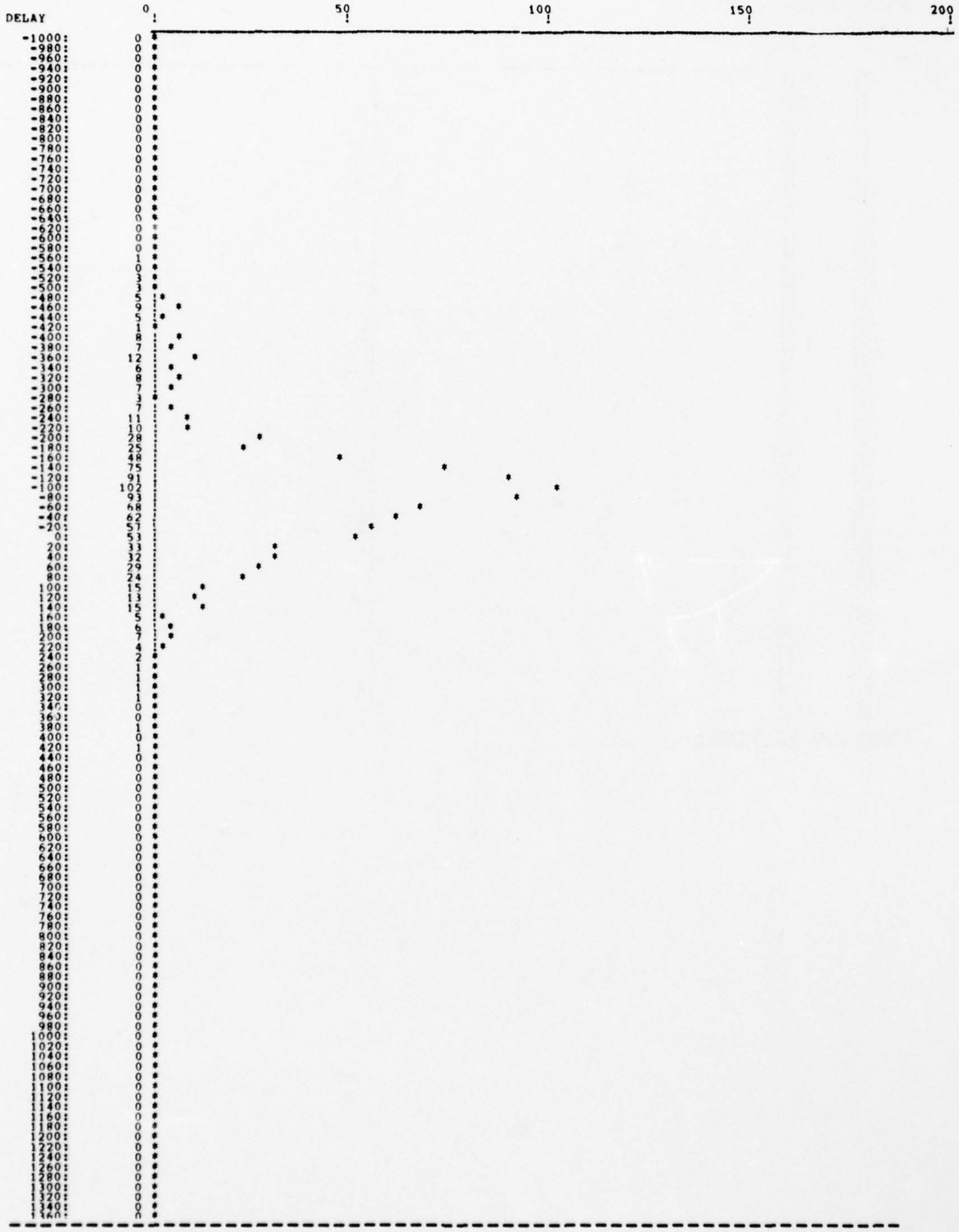
Relative Delays of DEC197EVE Figure A.6



```
                    RELATIVE DELAYS (MILLISECONDS) OF DEC197EVE

PSMF FILE = [350,77]D197E
PASSWORD = PW
LOW BOUNDARY =          -500
HIGH BOUNDARY =          500
AVERAGING INTERVAL =      10

SECONDS
      3:         -97
      7:         -18
     11:         -48
     15:         -12
     18:         -69
     22:         -37
     27:         -61
     30:         -62
     34:         -67
     37:         -58
     41:         -87
     45:          -3
     48:         -32
     52:         -54
     56:         -42
     60:          -4
     63:         -52
     67:         -58
     71:           8
     75:         -32
     79:          15
     82:        -100
     86:         -50
     89:         -48
     93:         -97
     97:          10
    101:         -85
    105:         -35
    108:         -75
    112:         -41
    115:         -96
    119:         -59
    123:         -18
    126:        -103
    130:         -57
    134:         -89
    138:         -45
    141:        -108
    144:        -117
    148:         -39
    152:        -126
    155:         -13
    159:         -54
    163:         -69
    167:         -69
    171:         -87
    174:         -45
    178:           8
    181:         -28
    186:         -21
    190:        -117
    194:         -78
    197:        -126
    201:         -45
    204:         -85
    208:         -44
    211:         -90
    215:        -142
    219:        -115
    223:        -119
    226:         -51
    230:         -99
    234:         -64
    237:        -107
    241:         -91
    244:         -87
    248:          -1
    252:         -95
    256:         -57
    259:         -94
    262:        -106
    267:         -21
    271:        -115
    274:        -103
    278:         -77
    282:         -48
    286:        -122
    289:        -132
    293:        -108
    297:        -108
    300:        -150
    303:        -108
    307:         -26
    311:         -76
    315:         -91
    320:        -110
    324:        -142
    327:         -88
    331:        -103
    334:         -79
    338:         -74
    342:        -130
    345:        -141
    349:        -151
    353:         -88
    356:        -131
    360:         -92
    363:        -168
    367:        -179
MINIMUM VALUE ENCOUNTERED =       -541
MAXIMUM VALUE ENCOUNTERED =        429
COUNT=          999
MEAN =          -74
STANDARD DEVIATION =        131
```

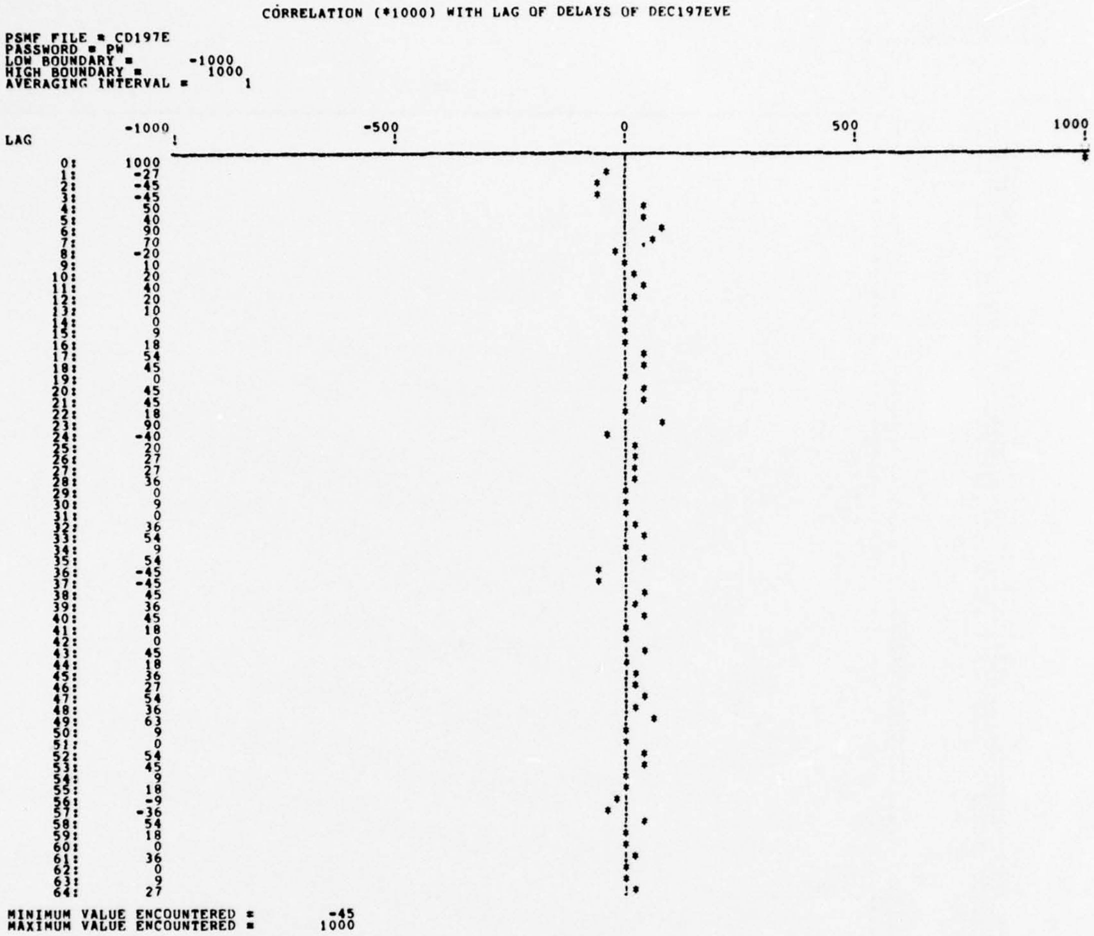
Histogram of Delay Times for DEC197EVE Figure A.7

```
                    HISTOGRAM OF DELAY TIMES FOR DEC197EVE

PSMF FILE = [350,77]HD197E
PASSWORD = PW
LOW BOUNDARY =            0
HIGH BOUNDARY =         200
AVERAGING INTERVAL =      1
```



```
                  0          50          100          150          200
DELAY             !           !            !            !            !

   -1000:      0
    -980:      0
    -960:      0
    -940:      0
    -920:      0
    -900:      0
    -880:      0
    -860:      0
    -840:      0
    -820:      0
    -800:      0
    -780:      0
    -760:      0
    -740:      0
    -720:      0
    -700:      0
    -680:      0
    -660:      0
    -640:      0
    -620:      0
    -600:      0
    -580:      0
    -560:      1
    -540:      0
    -520:      3
    -500:      3
    -480:      5
    -460:      9
    -440:      5
    -420:      1
    -400:      8
    -380:      7
    -360:     12
    -340:      6
    -320:      8
    -300:      7
    -280:      3
    -260:      7
    -240:     11
    -220:     10
    -200:     28
    -180:     25
    -160:     48
    -140:     75
    -120:     91
    -100:    102
     -80:     93
     -60:     68
     -40:     62
     -20:     57
       0:     53
      20:     33
      40:     32
      60:     29
      80:     24
     100:     15
     120:     13
     140:     15
     160:      5
     180:      6
     200:      7
     220:      4
     240:      2
     260:      1
     280:      1
     300:      1
     320:      1
     340:      0
     360:      0
     380:      1
     400:      0
     420:      1
     440:      0
     460:      0
     480:      0
     500:      0
     520:      0
     540:      0
     560:      0
     580:      0
     600:      0
     620:      0
     640:      0
     660:      0
     680:      0
     700:      0
     720:      0
     740:      0
     760:      0
     780:      0
     800:      0
     820:      0
     840:      0
     860:      0
     880:      0
     900:      0
     920:      0
     940:      0
     960:      0
     980:      0
    1000:      0
    1020:      0
    1040:      0
    1060:      0
    1080:      0
    1100:      0
    1120:      0
    1140:      0
    1160:      0
    1180:      0
    1200:      0
    1220:      0
    1240:      0
    1260:      0
    1280:      0
    1300:      0
    1320:      0
    1340:      0
    1360:      0
```

Correlation (1000) with Lag of Delays of DEC197EVEFigure A.8



```
                    CORRELATION (*1000) WITH LAG OF DELAYS OF DEC197EVE

PSMF FILE = CD197E
PASSWORD = PW
LOW BOUNDARY =        -1000
HIGH BOUNDARY =        1000
AVERAGING INTERVAL =      1

                  -1000            -500             0             500            1000
LAG                    !               !               !               !               !
        0:   1000
        1:    -27
        2:    -45
        3:    -45
        4:     50
        5:     40
        6:     90
        7:     70
        8:    -20
        9:     10
       10:     20
       11:     40
       12:     20
       13:     10
       14:      0
       15:      9
       16:     18
       17:     54
       18:     45
       19:      0
       20:     45
       21:     45
       22:     18
       23:     90
       24:    -40
       25:     20
       26:     27
       27:     27
       28:     36
       29:      0
       30:      9
       31:      0
       32:     36
       33:     54
       34:      9
       35:     54
       36:    -45
       37:    -45
       38:     45
       39:     36
       40:     45
       41:     18
       42:      0
       43:     45
       44:     18
       45:     36
       46:     27
       47:     54
       48:     36
       49:     63
       50:      9
       51:      0
       52:     54
       53:     45
       54:      9
       55:     18
       56:     -9
       57:    -36
       58:     54
       59:     18
       60:      0
       61:     36
       62:      0
       63:      9
       64:     27
MINIMUM VALUE ENCOUNTERED =        -45
MAXIMUM VALUE ENCOUNTERED =       1000
```

times lagged with relative delays of later packets. The horizontal axis is correlation coefficient times 1000, the vertical is lag. For example, the entry 28:36 means that the correlation coefficient of

relative delay of packet (k)   k=0, (Total - 28)
with
relative delay of packet (k+28)
is .036.

This measurement was designed to indicate any periodicity in network delay, as might be caused by queuing and buffering considerations. A significant correlation with 1000 degrees of freedom (99%) would be .074. Lags of 6 and 23 exceed this, but not radically so. It could be that the relatively low bit rate (~ 2.7 kbits/second) will not show buffering deficiencies.

B.

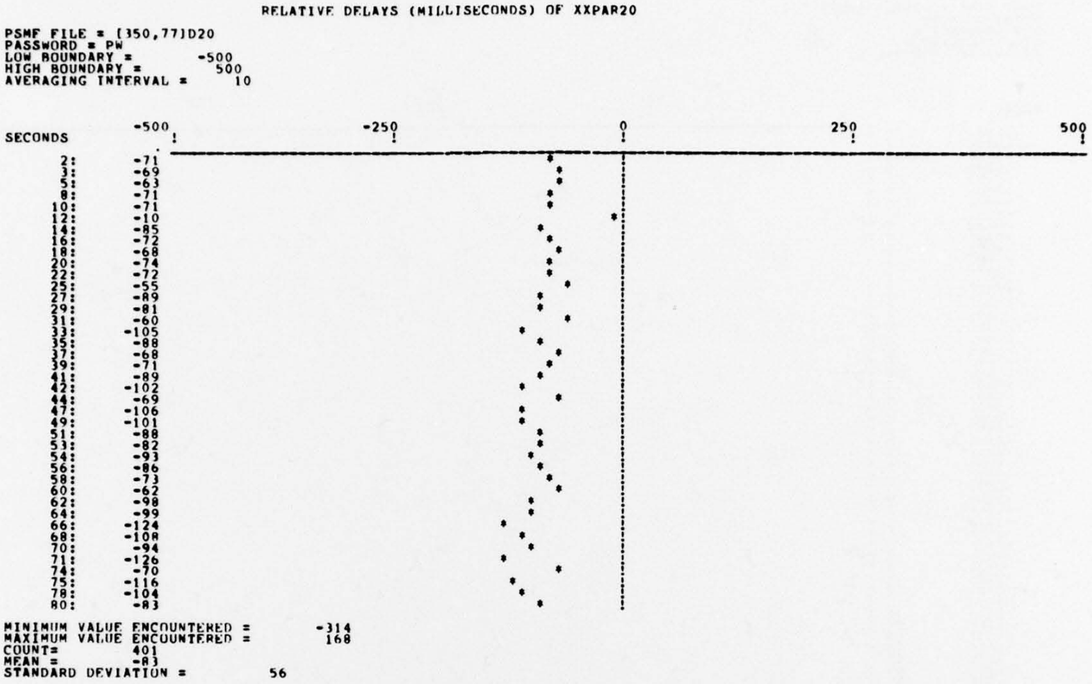
During January 1978, ISI researchers decided to investigate the relationship between network perturbations and packet size/frequency. Accordingly, a series of experiments were conducted to determine the nature of this relationship. This appendix examines three of these experiments. The measurement functions employed are the same as those demonstrated in Appendix A.

The three experiments dealt with messages containing three different packet size/frequencies.

Experiment	Packet size (*16 bits)	Frequency (/sec)
XXPAR20	29	5
XXPAR10	16	10
XXPAR5	10	20

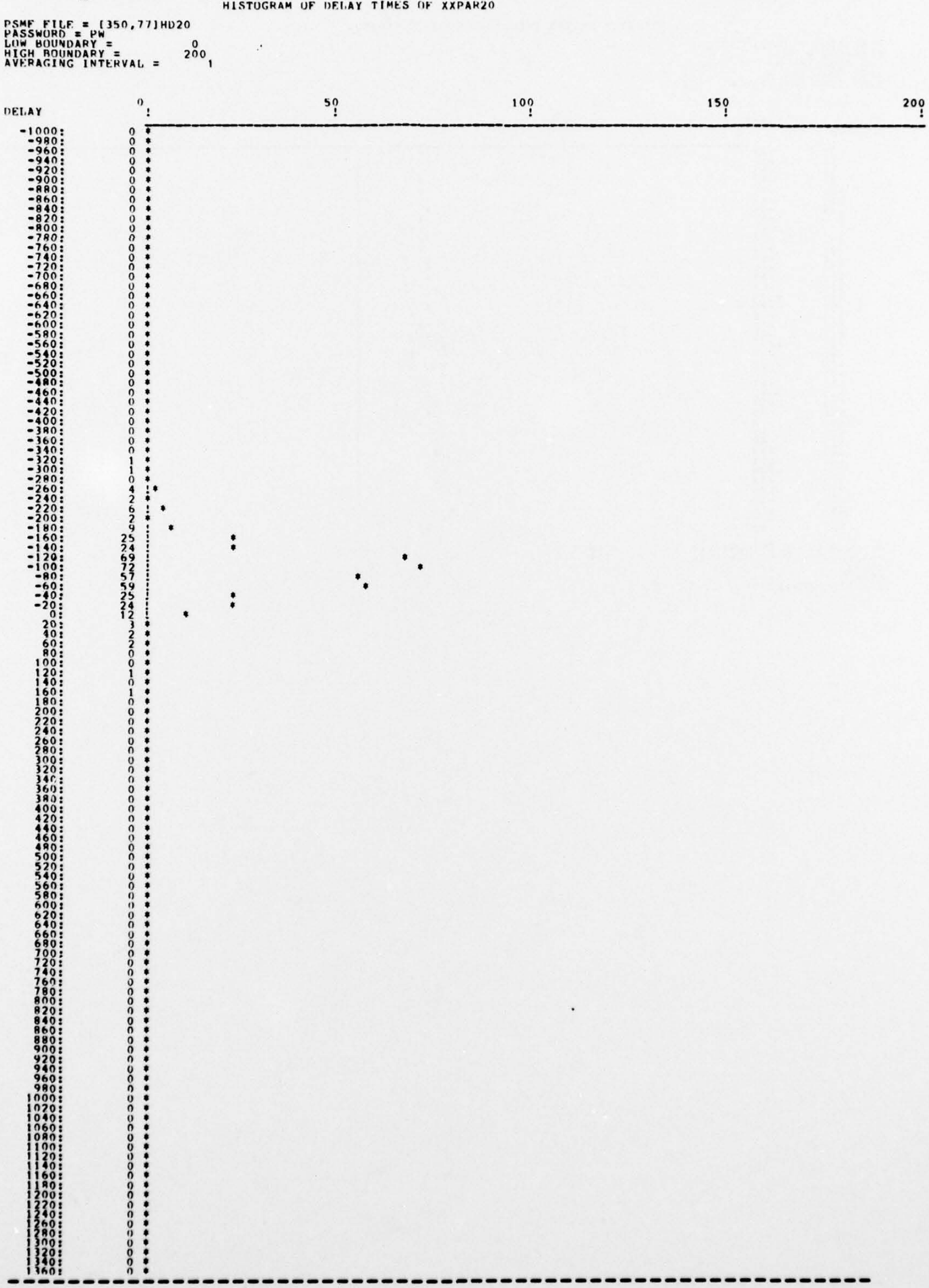
A graph of relative delays for experiment XXPAR20 is shown in Figure B1. It is a small file, but does not seem much different from that illustrated in Figure A6. The histogram of delays in Figure B2 is not unusual, and the correlation with lag measurement in Figure B3 is not very suggestive.

Relative Delays of XXPAR20 Figure B.1



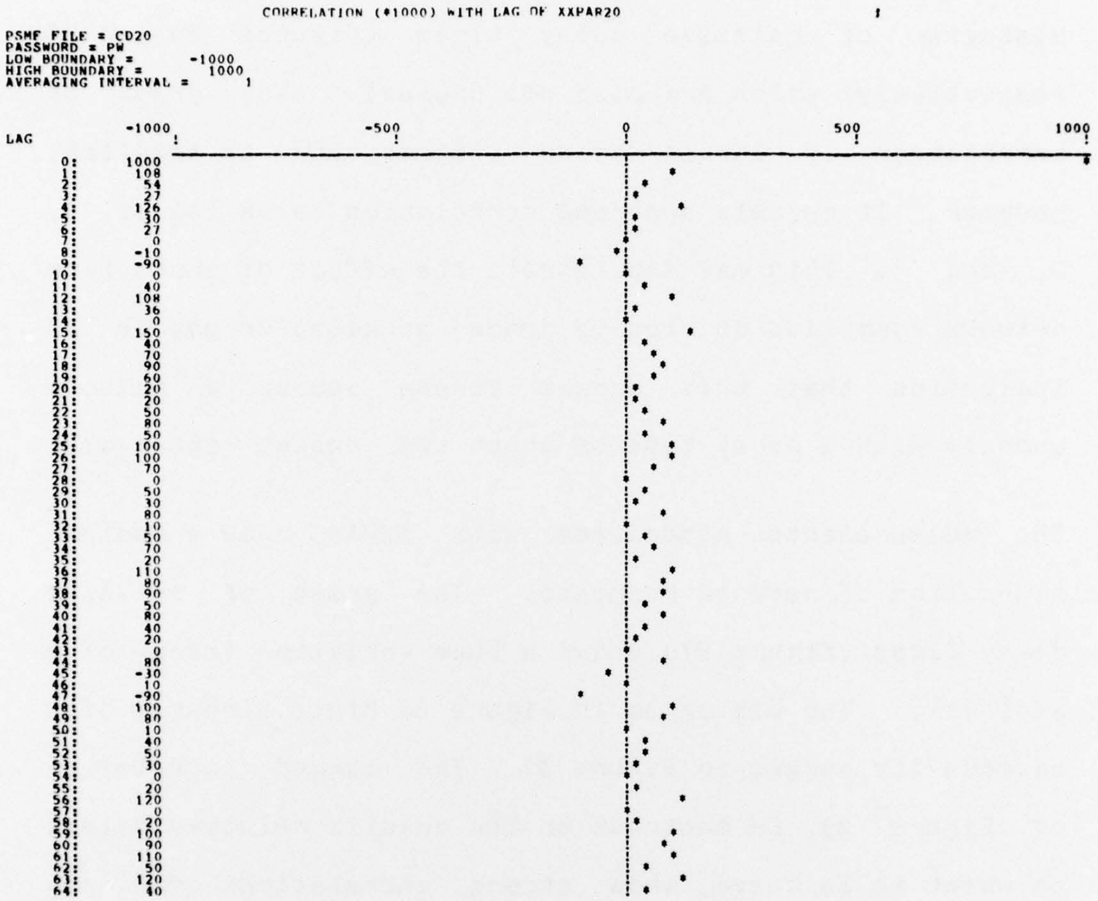
```
                              RELATIVE DELAYS (MILLISECONDS) OF XXPAR20
PSMF FILE = [350,77]D20
PASSWORD = PW
LOW BOUNDARY =        -500
HIGH BOUNDARY =        500
AVERAGING INTERVAL =    10

                -500            -250             0              250             500
SECONDS             !               !             !               !               !
        2:       -71
        3:       -69
        5:       -63
        8:       -71
       10:       -71
       12:       -10
       14:       -85
       16:       -72
       18:       -68
       20:       -74
       22:       -72
       25:       -55
       27:       -89
       29:       -81
       31:       -60
       33:      -105
       35:       -88
       37:       -68
       39:       -71
       41:       -89
       42:      -102
       44:       -69
       47:      -106
       49:      -101
       51:       -88
       53:       -82
       54:       -93
       56:       -86
       58:       -73
       60:       -62
       62:       -98
       64:       -99
       66:      -124
       68:      -108
       70:       -94
       71:      -126
       74:       -70
       75:      -116
       78:      -104
       80:       -83
MINIMUM VALUE ENCOUNTERED =        -314
MAXIMUM VALUE ENCOUNTERED =         168
COUNT=          401
MEAN =          -83
STANDARD DEVIATION =         56
```

Histogram of Delay Times of XXPAR20 Figure B.2



```
                              HISTOGRAM OF DELAY TIMES OF XXPAR20
PSMF FILE = [350,77]HD20
PASSWORD = PW
LOW BOUNDARY =          0
HIGH BOUNDARY =       200
AVERAGING INTERVAL =      1

DELAY          0                        50                       100                      150                      200
               !                        !                        !                        !                        !
               -----------------------------------------------------------------------------------------------------
-1000:       0 *
 -980:       0 *
 -960:       0 *
 -940:       0 *
 -920:       0 *
 -900:       0 *
 -880:       0 *
 -860:       0 *
 -840:       0 *
 -820:       0 *
 -800:       0 *
 -780:       0 *
 -760:       0 *
 -740:       0 *
 -720:       0 *
 -700:       0 *
 -680:       0 *
 -660:       0 *
 -640:       0 *
 -620:       0 *
 -600:       0 *
 -580:       0 *
 -560:       0 *
 -540:       0 *
 -520:       0 *
 -500:       0 *
 -480:       0 *
 -460:       0 *
 -440:       0 *
 -420:       0 *
 -400:       0 *
 -380:       0 *
 -360:       0 *
 -340:       0 *
 -320:       1 *
 -300:       1 *
 -280:       0 *
 -260:       4 ! *
 -240:       2 *
 -220:       6 !  *
 -200:       2 *
 -180:       9 !   *
 -160:      25 !           *
 -140:      24 !           *
 -120:      69 !                                 *
 -100:      72 !                                   *
  -80:      57 !                           *
  -60:      59 !                            *
  -40:      25 !           *
  -20:      24 !           *
    0:      12 !     *
   20:       3 *
   40:       2 *
   60:       2 *
   80:       0 *
  100:       0 *
  120:       1 *
  140:       0 *
  160:       1 *
  180:       0 *
  200:       0 *
  220:       0 *
  240:       0 *
  260:       0 *
  280:       0 *
  300:       0 *
  320:       0 *
  340:       0 *
  360:       0 *
  380:       0 *
  400:       0 *
  420:       0 *
  440:       0 *
  460:       0 *
  480:       0 *
  500:       0 *
  520:       0 *
  540:       0 *
  560:       0 *
  580:       0 *
  600:       0 *
  620:       0 *
  640:       0 *
  660:       0 *
  680:       0 *
  700:       0 *
  720:       0 *
  740:       0 *
  760:       0 *
  780:       0 *
  800:       0 *
  820:       0 *
  840:       0 *
  860:       0 *
  880:       0 *
  900:       0 *
  920:       0 *
  940:       0 *
  960:       0 *
  980:       0 *
 1000:       0 *
 1020:       0 *
 1040:       0 *
 1060:       0 *
 1080:       0 *
 1100:       0 *
 1120:       0 *
 1140:       0 *
 1160:       0 *
 1180:       0 *
 1200:       0 *
 1220:       0 *
 1240:       0 *
 1260:       0 *
 1280:       0 *
 1300:       0 *
 1320:       0 *
 1340:       0 *
 1360:       0 *
```

Correlation (1000) with Lag of XXPAR20   Figure B.3



```
                    CORRELATION (*1000) WITH LAG OF XXPAR20                    !

PSMF FILE = CD20
PASSWORD = PW
LOW BOUNDARY =        -1000
HIGH BOUNDARY =        1000
AVERAGING INTERVAL =      1

              -1000          -500            0            500          1000
LAG               !             !            !             !             !

       0:     1000
       1:      108
       2:       54
       3:       27
       4:      126
       5:       30
       6:       27
       7:        0
       8:      -10
       9:      -90
      10:        0
      11:       40
      12:      108
      13:       36
      14:        0
      15:      150
      16:       40
      17:       70
      18:       90
      19:       60
      20:       20
      21:       20
      22:       50
      23:       80
      24:       50
      25:      100
      26:      100
      27:       70
      28:        0
      29:       50
      30:       30
      31:       80
      32:       10
      33:       50
      34:       70
      35:       20
      36:      110
      37:       80
      38:       90
      39:       50
      40:       80
      41:       40
      42:       20
      43:        0
      44:       80
      45:      -30
      46:       10
      47:      -90
      48:      100
      49:       80
      50:       10
      51:       40
      52:       50
      53:       30
      54:        0
      55:       20
      56:      120
      57:        0
      58:       20
      59:      100
      60:       90
      61:      110
      62:       50
      63:      120
      64:       40
```

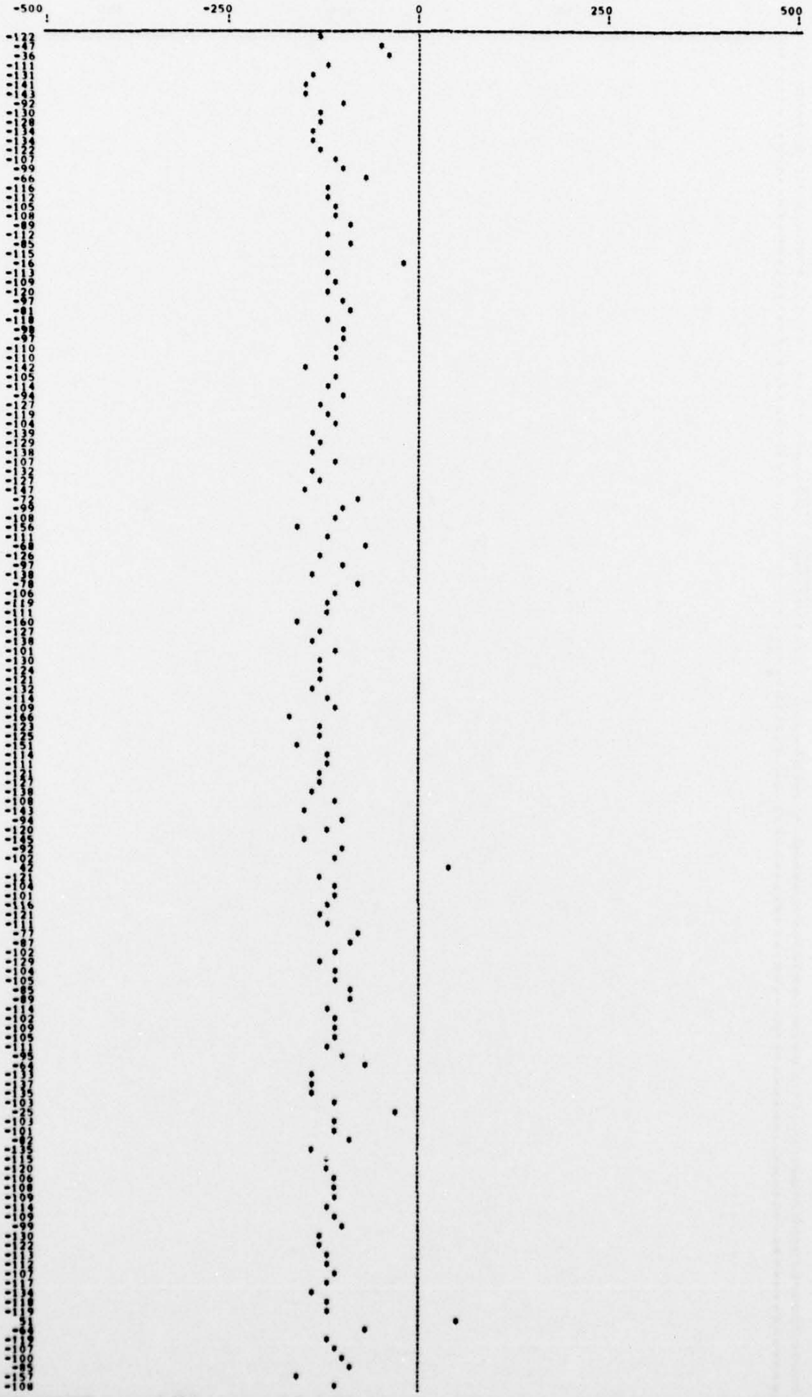
Corresponding measurements for XXPAR20 show a graph and histogram of relative delay times (Figures B4 and B5 respectively) which are also not unusual. The graph of correlation of lagged delay (Figure B6) is peculiar, however. It reveals a strong correlation for a lag of 1, 2, and 3. This may demonstrate the effect of short term network anomalies on closely spaced packets, or may be an indication that this packet stream caused a network anomaly with a decay time of about two packet intervals.

The measurements associated with XXPAR5 show a radical alteration of network response. The graph of relative delay times (Figure B7) shows a huge variation (range of 2 seconds). The histogram in Figure B8 hints strongly of a bi-modality suggested Figure A7. The lagged correlation of Figure B9, in contrast to the chaotic relative delays on which it is based, show strong correlations for all lags tested. A comparison with data represented in Figure B6 show a non-linear relationship between this correlation and packet frequency. There is every indication that this experiment produced a heavy drain on the network resources.

Again, however, no periodicity is seen in Figure B9. A glance at the count of missing and out-of-order messages in Figure B10 might reveal why. The measurement of lagged

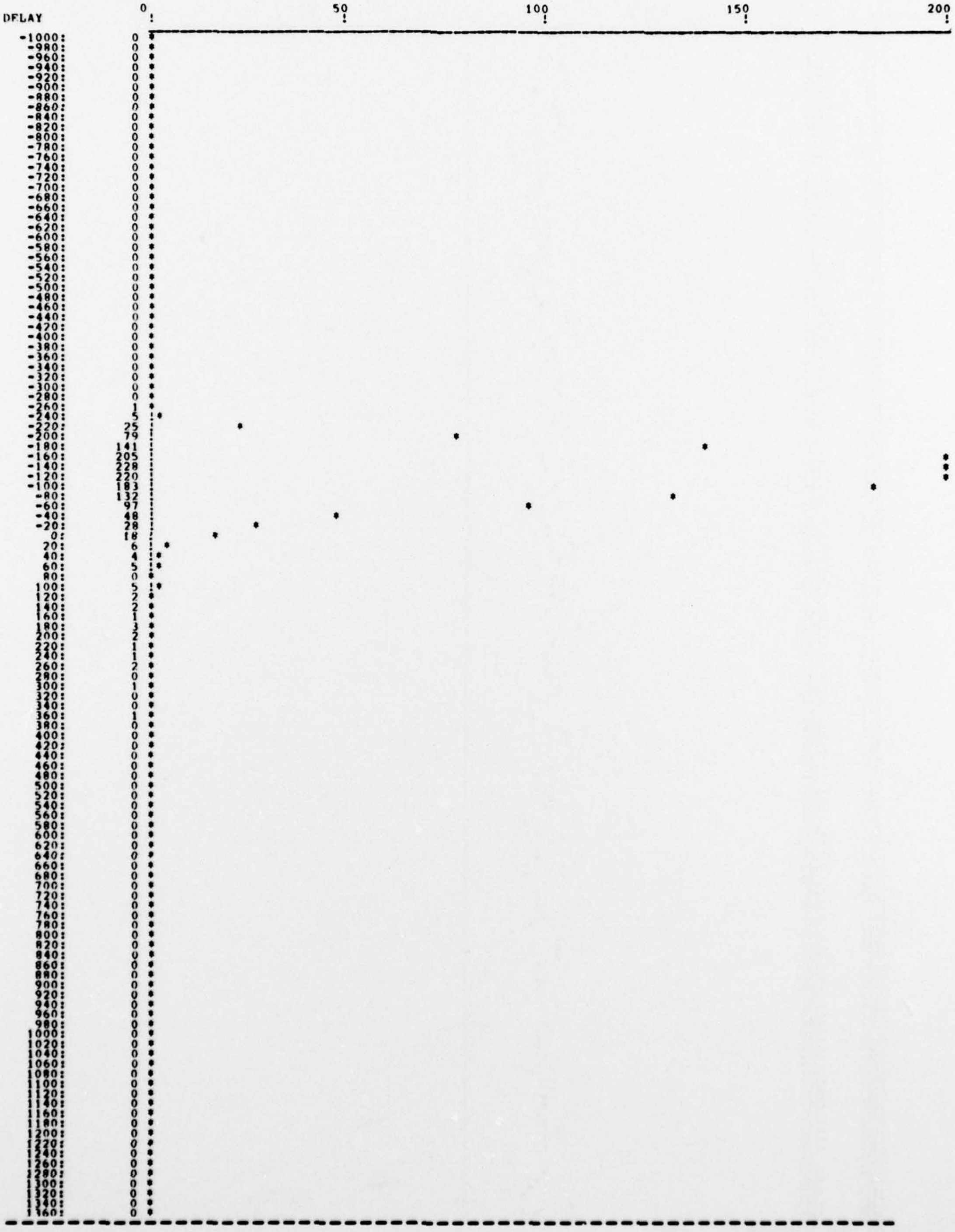
Relative Delays in XXPAR10 Figure B.4

```
                         RELATIVE DELAYS (MILLISECONDS) OF XXPAR10

PSMF FILE = [350,77]D10
PASSWORD = PW
LOW BOUNDARY =          -500
HIGH BOUNDARY =          500
AVERAGING INTERVAL =      10

SECONDS
      1:      -122
      2:       -47
      3:       -36
      4:      -111
      5:      -131
      6:      -141
      7:      -143
      8:       -92
      9:      -130
     10:      -128
     11:      -134
     13:      -134
     14:      -122
     15:      -107
     16:       -99
     16:       -66
     18:      -116
     19:      -112
     20:      -105
     21:      -108
     22:       -89
     24:      -112
     25:       -85
     26:      -115
     27:       -16
     28:      -113
     30:      -109
     31:      -120
     32:       -97
     33:       -81
     34:      -118
     35:       -98
     36:       -97
     37:      -110
     38:      -110
     38:      -142
     39:      -105
     40:      -114
     41:       -94
     42:      -127
     43:      -119
     44:      -104
     45:      -139
     46:      -129
     47:      -138
     48:      -107
     49:      -132
     50:      -127
     51:      -147
     52:       -72
     53:       -99
     54:      -108
     55:      -156
     56:      -111
     57:       -68
     58:      -126
     59:       -97
     59:      -138
     60:       -78
     61:      -106
     62:      -119
     63:      -111
     64:      -160
     65:      -127
     66:      -138
     67:      -101
     68:      -130
     69:      -124
     70:      -121
     71:      -132
     72:      -114
     73:      -109
     74:      -166
     75:      -123
     76:      -125
     77:      -151
     78:      -114
     78:      -111
     80:      -121
     81:      -127
     82:      -138
     83:      -108
     84:      -143
     85:       -94
     86:      -120
     87:      -145
     88:       -92
     89:      -102
     90:        42
     91:      -121
     92:      -104
     93:      -101
     94:      -116
     94:      -121
     95:      -111
     97:       -77
     97:       -87
     98:      -102
     99:      -129
    100:      -104
    101:      -105
    102:       -85
    103:       -89
    104:      -114
    105:      -102
    106:      -109
    107:      -105
    108:      -111
    110:       -95
    110:       -64
    111:      -133
    112:      -137
    113:      -135
    114:      -103
    115:       -25
    116:      -103
    118:      -101
    119:       -82
    120:      -135
    121:      -115
    122:      -120
    123:      -106
    124:      -108
    124:      -109
    125:      -114
    126:      -109
    127:       -99
    128:      -130
    129:      -122
    130:      -113
    131:      -112
    132:      -107
    133:      -117
    134:      -134
    135:      -118
    136:      -119
    137:        51
    138:       -64
    139:      -119
    140:      -107
    141:      -100
    142:       -85
    143:      -157
    143:      -108
MINIMUM VALUE ENCOUNTERED =          -258
MAXIMUM VALUE ENCOUNTERED =           366
COUNT=          1446
MEAN =          -109
STANDARD DEVIATION =          62
```



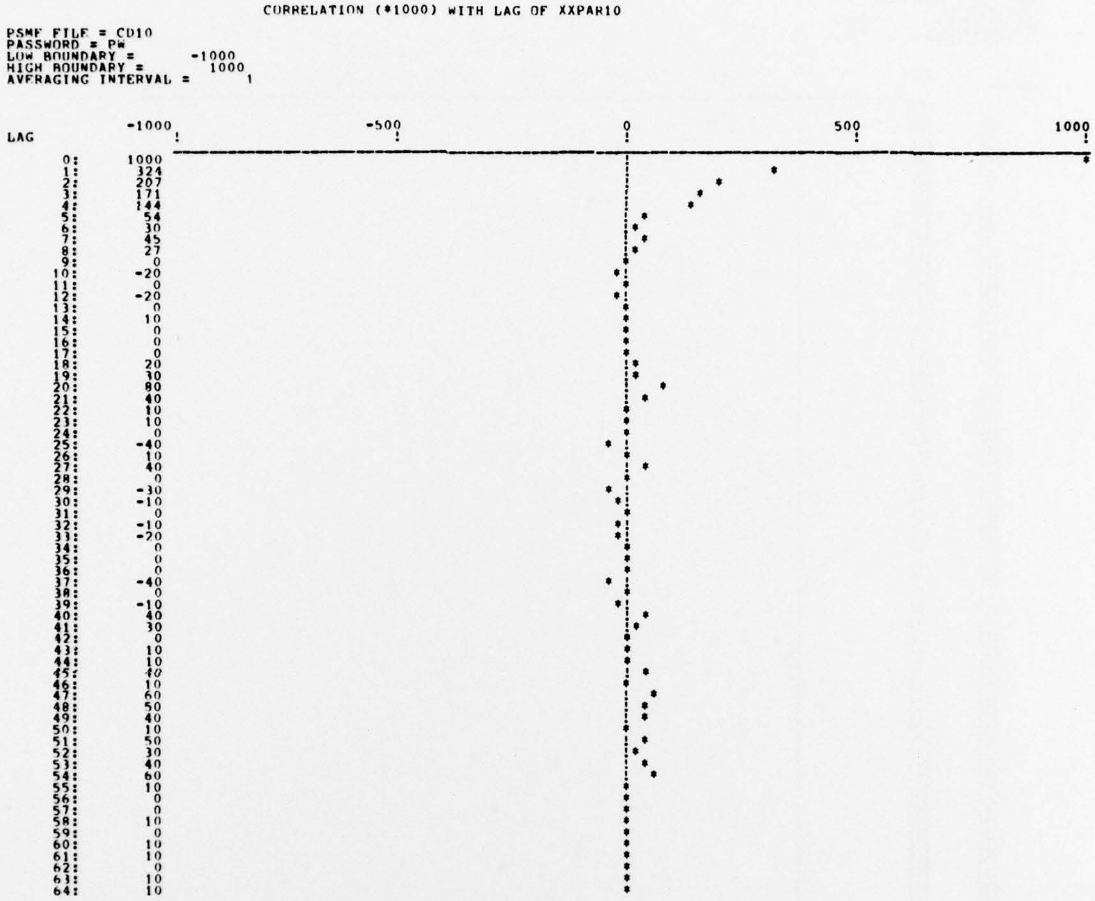
Histogram of Delay Times of XXPAR10 Figure B.5

```
                    HISTOGRAM OF DELAY TIMES OF XXPAR10

PSMF FILE = [350,77]HD10
PASSWORD = PW
LOW BOUNDARY =          0
HIGH BOUNDARY =       200
AVERAGING INTERVAL =     1
```



```
 DELAY
 -1000:      0
  -980:      0
  -960:      0
  -940:      0
  -920:      0
  -900:      0
  -880:      0
  -860:      0
  -840:      0
  -820:      0
  -800:      0
  -780:      0
  -760:      0
  -740:      0
  -720:      0
  -700:      0
  -680:      0
  -660:      0
  -640:      0
  -620:      0
  -600:      0
  -580:      0
  -560:      0
  -540:      0
  -520:      0
  -500:      0
  -480:      0
  -460:      0
  -440:      0
  -420:      0
  -400:      0
  -380:      0
  -360:      0
  -340:      0
  -320:      0
  -300:      0
  -280:      0
  -260:      1
  -240:      5
  -220:     25
  -200:     79
  -180:    141
  -160:    205
  -140:    228
  -120:    220
  -100:    183
   -80:    132
   -60:     97
   -40:     48
   -20:     28
     0:     18
    20:      6
    40:      4
    60:      5
    80:      0
   100:      5
   120:      2
   140:      2
   160:      1
   180:      3
   200:      2
   220:      1
   240:      1
   260:      2
   280:      0
   300:      1
   320:      0
   340:      0
   360:      1
   380:      0
   400:      0
   420:      0
   440:      0
   460:      0
   480:      0
   500:      0
   520:      0
   540:      0
   560:      0
   580:      0
   600:      0
   620:      0
   640:      0
   660:      0
   680:      0
   700:      0
   720:      0
   740:      0
   760:      0
   780:      0
   800:      0
   820:      0
   840:      0
   860:      0
   880:      0
   900:      0
   920:      0
   940:      0
   960:      0
   980:      0
  1000:      0
  1020:      0
  1040:      0
  1060:      0
  1080:      0
  1100:      0
  1120:      0
  1140:      0
  1160:      0
  1180:      0
  1200:      0
  1220:      0
  1240:      0
  1260:      0
  1280:      0
  1300:      0
  1320:      0
  1340:      0
  1360:      0
```

Correlation (1000) with Lag of XXPAR10 Figure B.6

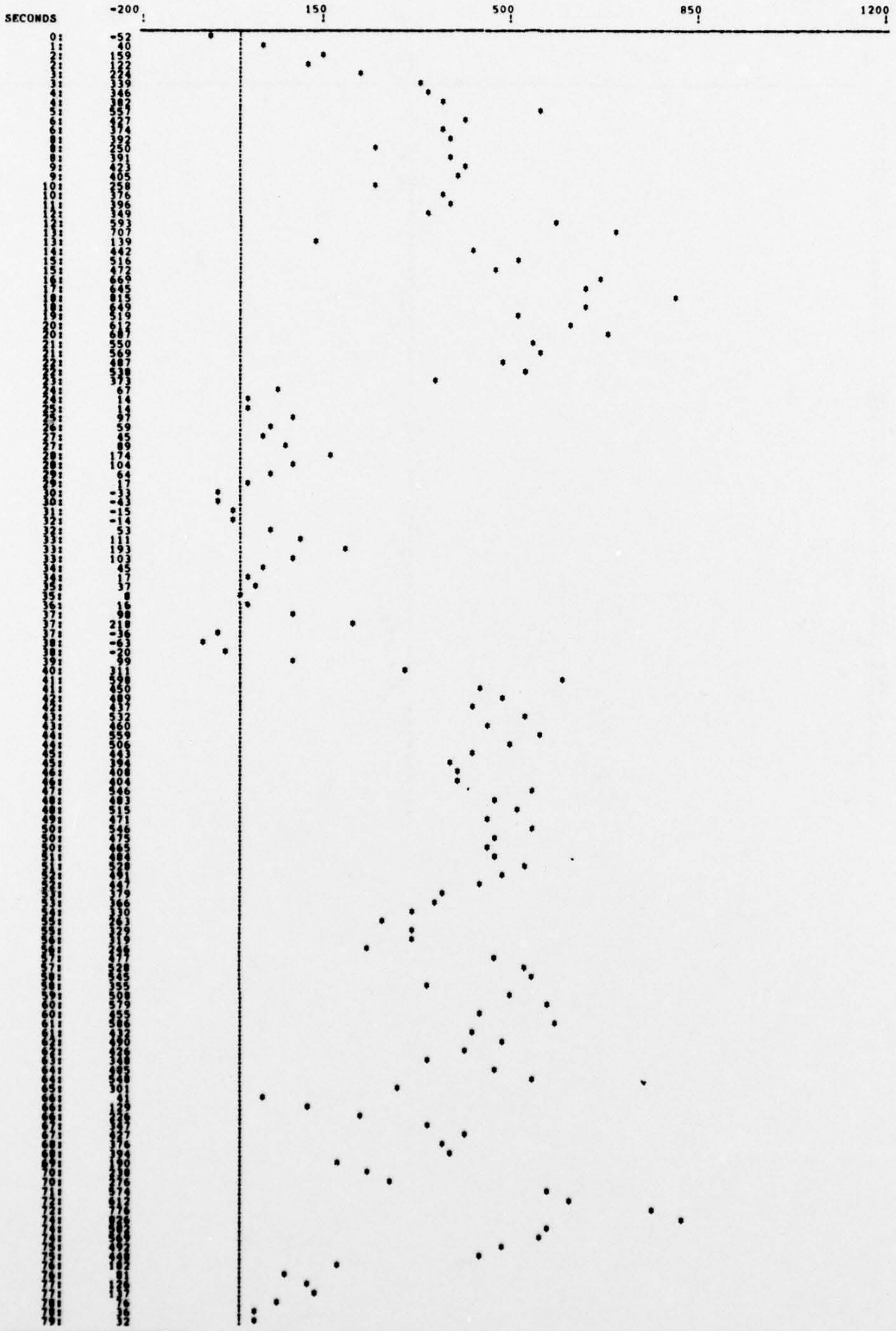


```
                              CORRELATION (*1000) WITH LAG OF XXPAR10

PSMF FILE = CD10
PASSWORD = PW
LOW BOUNDARY =          -1000
HIGH BOUNDARY =          1000
AVERAGING INTERVAL =        1

            -1000         -500          0           500          1000
LAG              !             !            !             !             !

       0:     1000
       1:      324
       2:      207
       3:      171
       4:      144
       5:       54
       6:       30
       7:       45
       8:       27
       9:        0
      10:      -20
      11:        0
      12:      -20
      13:        0
      14:       10
      15:        0
      16:        0
      17:        0
      18:       20
      19:       30
      20:       80
      21:       40
      22:       10
      23:       10
      24:        0
      25:      -40
      26:       10
      27:       40
      28:        0
      29:      -30
      30:      -10
      31:        0
      32:      -10
      33:      -20
      34:        0
      35:        0
      36:        0
      37:      -40
      38:        0
      39:      -10
      40:       40
      41:       30
      42:        0
      43:       10
      44:       10
      45:       40
      46:       10
      47:       60
      48:       50
      49:       40
      50:       10
      51:       50
      52:       30
      53:       40
      54:       60
      55:       10
      56:        0
      57:        0
      58:       10
      59:        0
      60:       10
      61:       10
      62:        0
      63:       10
      64:       10
```

Relative Delay of XXPAR5

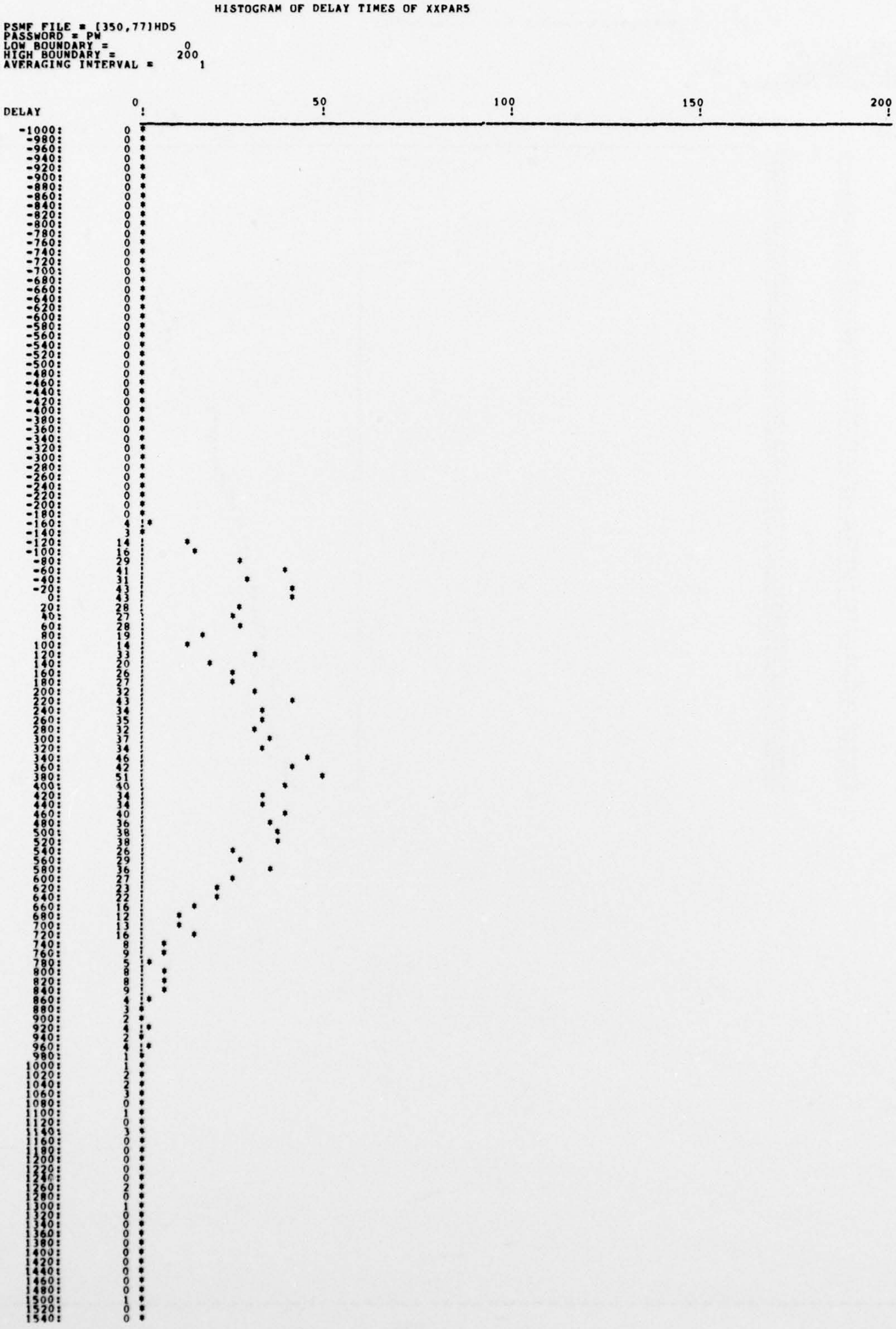
Figure B.7

```
                          RELATIVE DELAYS (MILLISECONDS) OF XXPAR5

PSMF FILE = [350,77]D5
PASSWORD = PW
LOW BOUNDARY =          -200
HIGH BOUNDARY =          1200
AVERAGING INTERVAL =       10
```



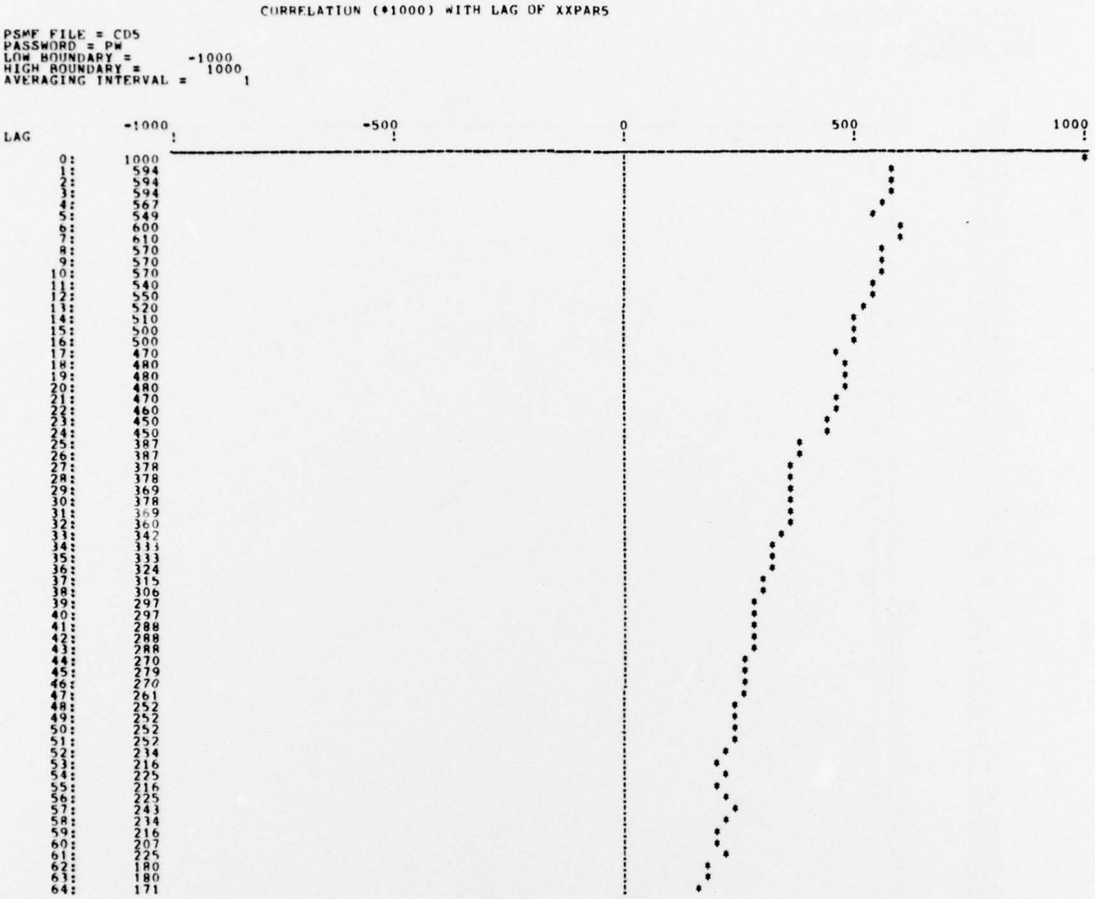
```
MINIMUM VALUE ENCOUNTERED =        -156
MAXIMUM VALUE ENCOUNTERED =        1503
COUNT=          1395
MEAN =           334
STANDARD DEVIATION =        276
```

Histogram of Delay Times of XXPAR5 — Figure B.8



```
                         HISTOGRAM OF DELAY TIMES OF XXPAR5

PSMF FILE = [350,77]HD5
PASSWORD = PW
LOW BOUNDARY =          0
HIGH BOUNDARY =       200
AVERAGING INTERVAL =      1

                0                       50                       100                      150                      200
DELAY           !                        !                        !                        !                        !
  -1000:      0 *
   -980:      0 *
   -960:      0 *
   -940:      0 *
   -920:      0 *
   -900:      0 *
   -880:      0 *
   -860:      0 *
   -840:      0 *
   -820:      0 *
   -800:      0 *
   -780:      0 *
   -760:      0 *
   -740:      0 *
   -720:      0 *
   -700:      0 *
   -680:      0 *
   -660:      0 *
   -640:      0 *
   -620:      0 *
   -600:      0 *
   -580:      0 *
   -560:      0 *
   -540:      0 *
   -520:      0 *
   -500:      0 *
   -480:      0 *
   -460:      0 *
   -440:      0 *
   -420:      0 *
   -400:      0 *
   -380:      0 *
   -360:      0 *
   -340:      0 *
   -320:      0 *
   -300:      0 *
   -280:      0 *
   -260:      0 *
   -240:      0 *
   -220:      0 *
   -200:      0 *
   -180:      0 *
   -160:      4 !*
   -140:      3 *
   -120:     14 !     *
   -100:     16 !      *
    -80:     29 !            *
    -60:     41 !                  *
    -40:     31 !             *
    -20:     43 !                   *
      0:     43 !                   *
     20:     28 !            *
     40:     27 !           *
     60:     28 !            *
     80:     19 !       *
    100:     14 !     *
    120:     33 !              *
    140:     20 !        *
    160:     26 !           *
    180:     27 !           *
    200:     32 !              *
    220:     43 !                   *
    240:     34 !               *
    260:     35 !               *
    280:     32 !              *
    300:     37 !                *
    320:     34 !               *
    340:     46 !                     *
    360:     42 !                   *
    380:     51 !                       *
    400:     40 !                  *
    420:     34 !               *
    440:     34 !               *
    460:     40 !                  *
    480:     36 !                *
    500:     38 !                 *
    520:     38 !                 *
    540:     26 !           *
    560:     29 !            *
    580:     36 !                *
    600:     27 !           *
    620:     23 !         *
    640:     22 !         *
    660:     16 !      *
    680:     12 !    *
    700:     13 !    *
    720:     16 !      *
    740:      8 !  *
    760:      9 !  *
    780:      5 !*
    800:      8 !  *
    820:      8 !  *
    840:      9 !  *
    860:      4 !*
    880:      3 *
    900:      2 *
    920:      4 !*
    940:      2 *
    960:      4 !*
    980:      1 *
   1000:      1 *
   1020:      2 *
   1040:      2 *
   1060:      3 *
   1080:      0 *
   1100:      1 *
   1120:      0 *
   1140:      3 *
   1160:      0 *
   1180:      0 *
   1200:      0 *
   1220:      0 *
   1240:      0 *
   1260:      2 *
   1280:      0 *
   1300:      1 *
   1320:      0 *
   1340:      0 *
   1360:      0 *
   1380:      0 *
   1400:      0 *
   1420:      0 *
   1440:      0 *
   1460:      0 *
   1480:      0 *
   1500:      1 *
   1520:      0 *
   1540:      0 *
```

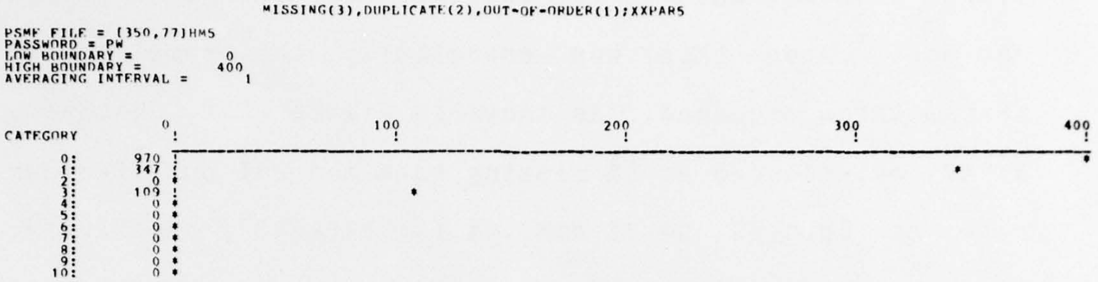
Correlation (1000) with Lag of XXPAR5   Figure B.9



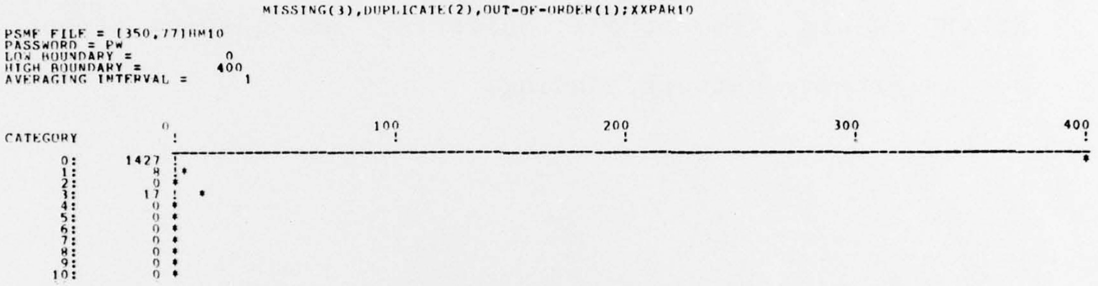
```
                    CORRELATION (*1000) WITH LAG OF XXPAR5

PSMF FILE = CD5
PASSWORD = PW
LOW BOUNDARY =        -1000
HIGH BOUNDARY =        1000
AVERAGING INTERVAL =      1

               -1000           -500            0            500          1000
LAG                 !             !            !             !             !

       0:     1000
       1:      594
       2:      594
       3:      594
       4:      567
       5:      549
       6:      600
       7:      610
       8:      570
       9:      570
      10:      570
      11:      540
      12:      550
      13:      520
      14:      510
      15:      500
      16:      500
      17:      470
      18:      480
      19:      480
      20:      480
      21:      470
      22:      460
      23:      450
      24:      450
      25:      387
      26:      387
      27:      378
      28:      378
      29:      369
      30:      378
      31:      369
      32:      360
      33:      342
      34:      333
      35:      333
      36:      324
      37:      315
      38:      306
      39:      297
      40:      297
      41:      288
      42:      288
      43:      288
      44:      270
      45:      279
      46:      270
      47:      261
      48:      252
      49:      252
      50:      252
      51:      252
      52:      234
      53:      216
      54:      225
      55:      216
      56:      225
      57:      243
      58:      234
      59:      216
      60:      207
      61:      225
      62:      180
      63:      180
      64:      171
```

correlation was made in the sequence of packet arrival. In most cases this was essentially the same as the transmission sequence. As shown in Figure B10, however, XXPAR5 experienced an 8% missing rate and 24% out-of-order rate as opposed to 1% and .6% for XXPAR10 respectively. Perhaps a study of lagged correlation on a reconstructed XXPAR5 would demonstrate buffering and queuing effects during extreme network loading.

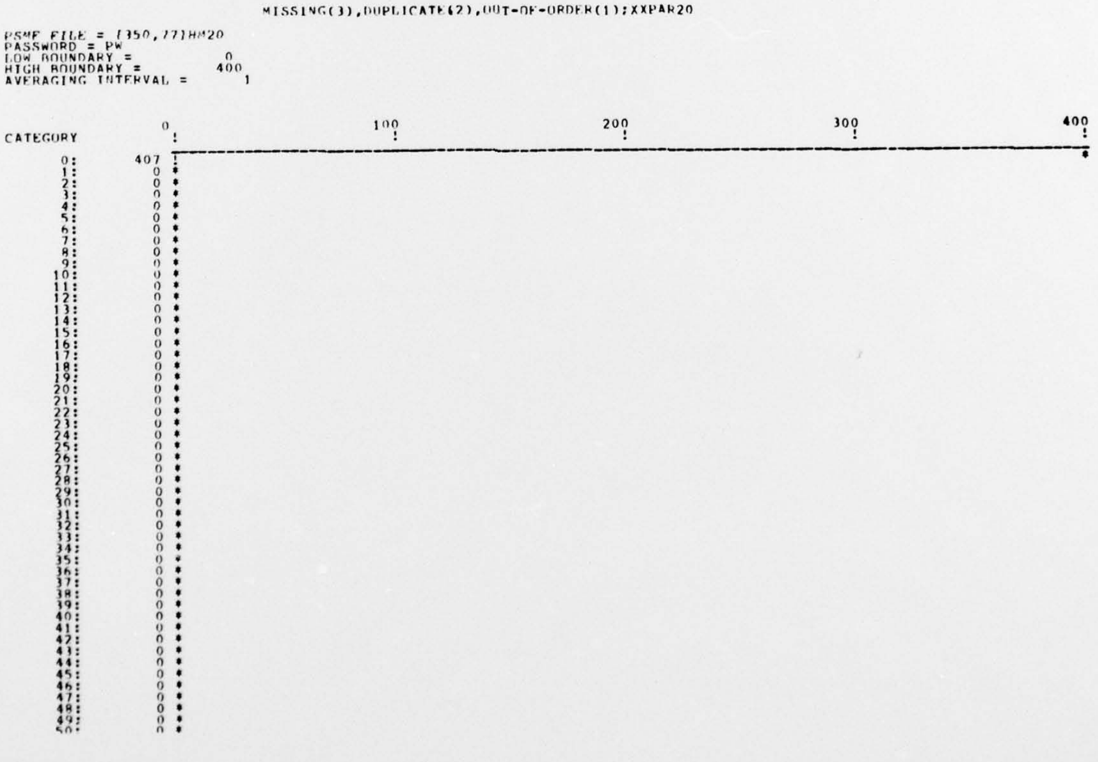
Figure B.10



```
                              MISSING(3),DUPLICATE(2),OUT-OF-ORDER(1);XXPAR5

PSMF FILE = [350,77]HM5
PASSWORD = PW
LOW BOUNDARY =           0
HIGH BOUNDARY =        400
AVERAGING INTERVAL =     1

                      0                        100                      200                      300                      400
CATEGORY               !                        !                        !                        !                        !
                       -----------------------------------------------------------------------------------------------------
      0:           970 !                                                                                                   *
      1:           347 !                                                                                      *
      2:             0 *
      3:           109 !                          *
      4:             0 *
      5:             0 *
      6:             0 *
      7:             0 *
      8:             0 *
      9:             0 *
     10:             0 *
```



```
                              MISSING(3),DUPLICATE(2),OUT-OF-ORDER(1);XXPAR10

PSMF FILE = [350,77]HM10
PASSWORD = PW
LOW BOUNDARY =           0
HIGH BOUNDARY =        400
AVERAGING INTERVAL =     1

                      0                        100                      200                      300                      400
CATEGORY               !                        !                        !                        !                        !
                       -----------------------------------------------------------------------------------------------------
      0:          1427 !                                                                                                   *
      1:             8 ! *
      2:             0 *
      3:            17 !   *
      4:             0 *
      5:             0 *
      6:             0 *
      7:             0 *
      8:             0 *
      9:             0 *
     10:             0 *
```



```
                              MISSING(3),DUPLICATE(2),OUT-OF-ORDER(1);XXPAR20

PSMF FILE = [350,77]HM20
PASSWORD = PW
LOW BOUNDARY =           0
HIGH BOUNDARY =        400
AVERAGING INTERVAL =     1

                      0                        100                      200                      300                      400
CATEGORY               !                        !                        !                        !                        !
                       -----------------------------------------------------------------------------------------------------
      0:           407 !                                                                                                   *
      1:             0 *
      2:             0 *
      3:             0 *
      4:             0 *
      5:             0 *
      6:             0 *
      7:             0 *
      8:             0 *
      9:             0 *
     10:             0 *
     11:             0 *
     12:             0 *
     13:             0 *
     14:             0 *
     15:             0 *
     16:             0 *
     17:             0 *
     18:             0 *
     19:             0 *
     20:             0 *
     21:             0 *
     22:             0 *
     23:             0 *
     24:             0 *
     25:             0 *
     26:             0 *
     27:             0 *
     28:             0 *
     29:             0 *
     30:             0 *
     31:             0 *
     32:             0 *
     33:             0 *
     34:             0 *
     35:             0 *
     36:             0 *
     37:             0 *
     38:             0 *
     39:             0 *
     40:             0 *
     41:             0 *
     42:             0 *
     43:             0 *
     44:             0 *
     45:             0 *
     46:             0 *
     47:             0 *
     48:             0 *
     49:             0 *
     50:             0 *
```

References

[COHEN]

Cohen, D. "Specifications for the Network Voice Protocol", ISI/RR-75-39, Information Sciences Institute, University of Southern California, March 1976.

[COHEN 2]

Cohen, D. "The Network Voice Conference Protocol (NVCP)", Information Sciences Institute, University of Southern California, February 1976.

[LOW]

Low, D. "Packet Speech Measurement Facility: Semi-Annual Technical Report I", Technical Report No. CCA-77-07, Computer Corporation of America, 575 Technology Square, Cambridge, Massachusetts, 02139, July, 1977.